Convolution-Based Interpolation for Fast, High-Quality Rotation of Images

Michael Unser, *Senior Member, IEEE*, Philippe Thévenaz, and Leonid Yaroslavsky

Abstract—**This paper focuses on the design of fast algorithms for rotating images and preserving high quality. The basis for the approach is a decomposition of a rotation into a sequence of one-dimensional translations. As the accuracy of these operations is critical, we introduce a general theoretical framework that addresses their design and performance. We also investigate the issue of optimality and present an improved least-square formulation of the problem. This approach leads to a separable three-pass implementation of a rotation using one-dimensional convolutions only. We provide explicit filter formulas for several continuous signal models including spline and bandlimited representations. Finally, we present rotation experiments and compare the currently standard techniques with the various versions of our algorithm. Our results indicate that the present algorithm in its higher-order versions outperforms all standard high-accuracy methods of which we are aware, both in terms of speed and quality. Its computational complexity increases linearly with the order of accuracy. The best-quality results are obtained with the sinc-based algorithm, which can be implemented using simple one-dimensional FFT's.**

I. Introduction

THERE are certain applications, such as radiology or digital photography, that require images to be rotated with the greatest possible care in order to preserve the integrity of the data. As it turns out, the quality of a rotation algorithm is directly related to the order of accuracy (to be defined in Section II) of the underlying interpolation model. Zero and first order approaches (nearest neighbor and bilinear interpolation) are extremely simple to implement but tend to produce images with noticeable artifacts (blocking or smoothing) [1]. More satisfactory results can be obtained using small-kernel cubic convolution techniques (piecewise cubic model) [2]–[4], or bicubic spline interpolation [5], which is nonlocal but still manageable computationally because of the availability of fast algorithms [6]. Unfortunately, such nonseparable methods become impractical for higher-order models because of the increase of the size of the local neighborhood that needs to be considered and the complexity of the corresponding 2-D interpolation formulas.

Manuscript received June 2, 1994; revised January 20, 1995. The associate editor coordinating the review of this paper and approving it for publication was Prof. Charles A. Bouman.

The authors are with the Biomedical Engineering and Instrumentation Program, National Center for Research Resources, National Institutes of Health, Bethesda, MD 20892-5766 USA.

IEEE Log Number 9413837.

One way to get around this problem is to decompose the rotation into a sequence of 1-D transformations along the x and y directions, which leads to a separable implementation. This makes it possible to use signal models of much higher order as in the standard (nonseparable) case, while at the same time reducing the amount of computation. There are several such decompositions that have been described in the literature and that can provide the starting point for a high-quality rotation algorithm.

Two-pass algorithms are typically based on the following decomposition of a rotation matrix [7]–[9]:

$$\boldsymbol{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \tan\theta & 1/\cos\theta \end{bmatrix} \times \begin{bmatrix} \cos\theta & -\sin\theta \\ 0 & 1 \end{bmatrix} \tag{1}$$

where the first and second terms represent 1-D shearing and expansion/contraction transformations along the y- and x-axes, respectively. This particular factorization can be advantageously exploited for hardware implementations [10], including DSP. It is also well-suited for dealing with very large images, since the data need to be accessed only one row (or column) at a time [11]. Its main drawback is that it requires some intermediate signal contraction, which not only complicates the implementation but also introduces errors. Even if aliasing can be prevented by an appropriate spectral shaping, there is nonetheless a loss of high spatial frequencies that becomes more pronounced for larger angles. This effect is most detrimental at 45° since the image needs to be reduced by a factor $\sqrt{2}$. This observation was first made by Friedmann, who proposed to correct the problem by using an additional resampling pass to increase the sampling rate of the image before rotation [9]. This augmented version gives rise to a three-pass algorithm, but it is not the most efficient one because it still requires scaling.

In fact, there is a factorization that requires no scaling at all and that appears to have been discovered independently by several authors [12]–[14]. It is given by

$$\boldsymbol{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} 1 & -\tan\theta/2 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ \sin\theta & 1 \end{bmatrix} \times \begin{bmatrix} 1 & -\tan\theta/2 \\ 0 & 1 \end{bmatrix}. \tag{2}$$

The whole transformation can be decomposed in an appropriate sequence of 1-D signal translations that can all be

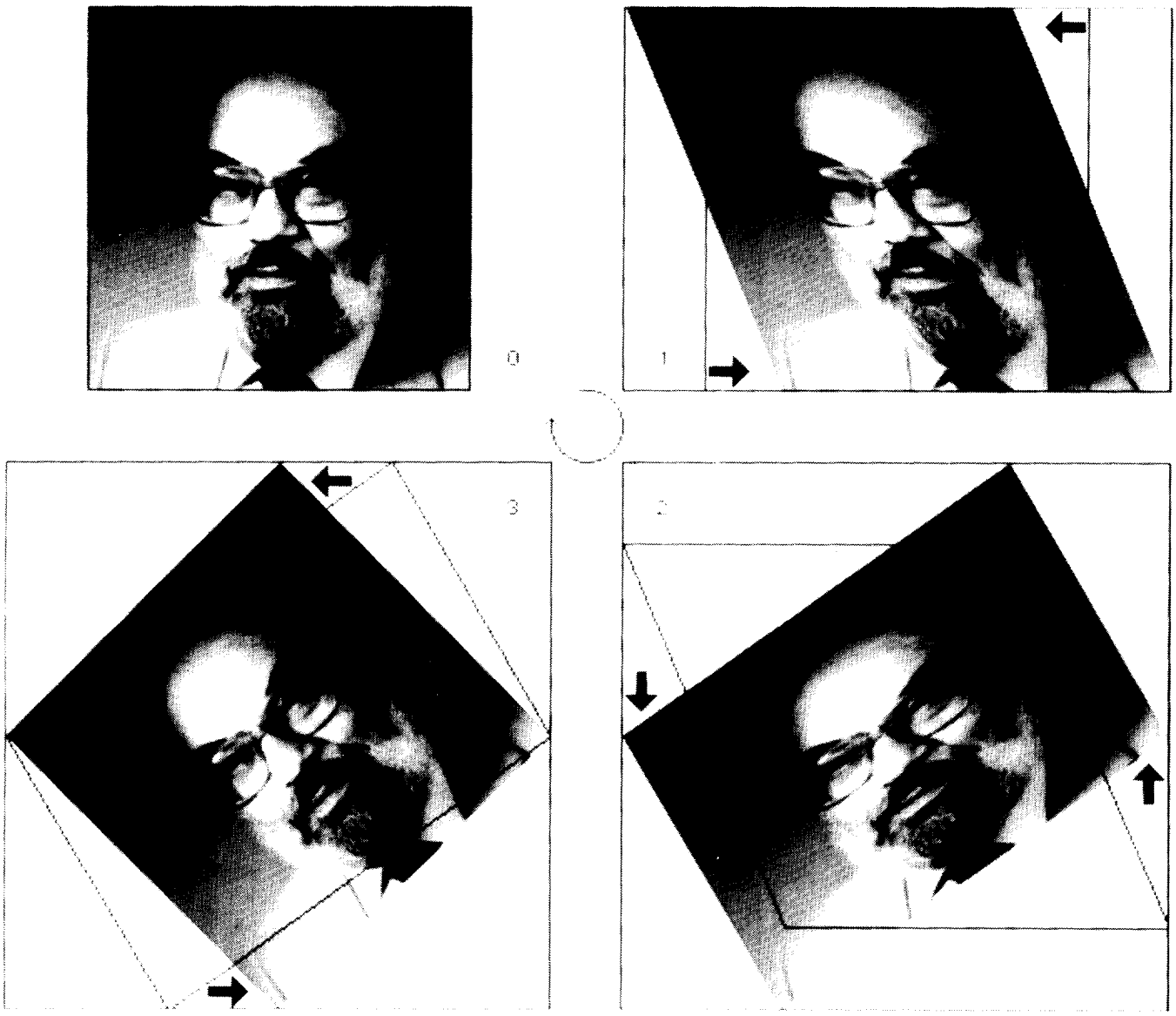


Fig. 1. Illustration of the three-pass rotation algorithm. (0) Original image; (1) shearing along the x-axis; (2) shearing along the y-axis; (3) final shearing along the x-axis, which yields the rotated image.

implemented via simple convolutions. It suggests the three-pass implementation of a rotation illustrated in Fig. 1. Each elementary transformation in (2) corresponds to a shearing of the image in the x or y direction. The processing is one-dimensional in nature since each row (resp., each column) is simply translated by an offset $\Delta_x = -y \cdot \tan\theta/2$ (resp., $\Delta_y = x \cdot \sin\theta$) that is proportional to its vertical (resp., horizontal) coordinate. The advantage of the second factorization over the first one is that the determinant of each of the elementary matrices in (2) is one, indicating that all areas in the image are preserved throughout the transformation process. This property suggests that this approach should better preserve higher spatial frequencies. Because the method requires no signal rescaling, it is much easier to implement.

For these reasons, we have chosen to base our approach on the second decomposition, which allows us to rotate an image using 1-D translations only. Our main focus in this paper will be the issue of rotation quality and the design of algorithms that are more accurate than those that have been previously reported. Most of the paper will be devoted to analyzing the proper way of translating a discrete one-dimensional signal, since this fundamental operation needs to be performed at least three times. For this purpose, we will introduce a general theoretical framework for the design and analysis of convolution-based interpolators. We will also relate the order of accuracy of an interpolator to the spectral characteristics of the underlying generating function.

For a given interpolation model, we will consider two different approaches for the design of translation operators. The first uses direct resampling, while the second minimizes the error between the ideal signal translation and its approximation within the given function space. After having dealt with these theoretical concerns, we will experimentally compare the outcome of the direct implementation of an image rotation

(that is, with full 2-D interpolation) versus the two- and three-pass implementations. In the process, we will try several types of interpolation schemes and provide a performance assessment.

A. Notations and Operators

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

$$\langle s(x), r(x) \rangle, = \int_{-\infty}^{+\infty} s(x) r(x)\, dx. \tag{3}$$

Two examples of L_2-functions that will be used are $\mathrm{rect}(x)$ (the symmetrical unit rectangular pulse) and $\mathrm{sinc}(x) = \sin(\pi x)/(\pi x)$.

l_2 is the vector space of square-summable sequences (or discrete signals) $a(k), k \in Z$. Square brackets are used to represent sequences that are obtained from the sampling of a L_2-function; i.e., $s[k] := s(x)|_{x=k}$. The convolution between two sequences a and b is denoted by $b * a(k)$. The sequence $b(k)$ can be viewed as a discrete convolution operator (or digital filter) that is applied to the signal $a \in l_2$, and is characterized by its transfer function $B(z) = \Sigma_{k \in Z}\, b(k) z^{-k}$. Such a filter defines an invertible convolution operator from l_2 into itself if and only if there exist two strictly positive constants m and M such that

$$m \leq |B(e^{j\omega})|^2 \leq M. \quad \text{a.e.} \tag{4}$$

This condition insures the existence and stability of the inverse filter, which we denote by

$$(b)^{-1}(k) \overset{z}{\leftrightarrow} 1/B(z). \tag{5}$$

II. Convolution-Based Interpolation

The important feature of the three-pass rotation procedure is that it involves one-dimensional translations only. Since the approach requires two intermediate steps, it is essential that the translations be performed with the greatest possible care. In order to address this issue, we introduce a general theoretical framework that is well-suited for the design and analysis of such operators. We also provide detailed implementation formulas for a variety of continuous signal models. The key practical result is that the corresponding translation operators can be computed by means of simple one-dimensional convolution operations (digital filter).

A. Design and Implementation of the Translation Operator

Let $s(x) \in L_2$ represent a function of the one-dimensional continuous variable x and $T_\Delta: L_2 \rightarrow L_2$ denote the one-dimensional translation operator by Δ; i.e., $T_\Delta s(x) = s(x - \Delta)$. If one starts with a sequence of samples $s[k] \in l_2$, the most natural way to implement T_Δ is to determine an interpolation model $s(x)$ such that $s[k] = s(x)|_{x=k}$ and then resample the translated version of this function. A general approach to constructing such interpolants is to consider the class of functions generated from the translates of a single function $\varphi(x)$ [15]. The corresponding function space $V(\varphi) \subset L_2$ is defined as

$$V(\varphi) = \left\{ s(x) = \sum_{k \in Z} c(k) \varphi(x - k) | c \in l_2 \right\}. \tag{6}$$

The only restriction on the choice of the *generating* function φ is that the set $\{\varphi(x - k)\}_{k \in Z}$ is a Riesz basis of $V(\varphi)$; this is equivalent to the condition

$$A \leq \sum_{k \in Z} |\Phi(\omega + 2\pi k)|^2 \leq B \quad \text{a.e.} \tag{7}$$

where $\Phi(\omega)$ is the Fourier transform of $\varphi(x)$, and where A and B are two strictly positive constants. This constraint insures that each function $s(x)$ in $V(\varphi)$ is uniquely characterized by the sequence of its coefficients $c(k)$.

Assuming that these coefficients are known, the sampled version of our translated signal can be evaluated through the following convolution

$$(T_\Delta s)[k] = \sum_{l \in Z} c(l) \varphi(k - \Delta - l) = b_\Delta * c(k) \tag{8}$$

where the digital filtering kernel b_Δ is the resampled version of $T_\Delta \varphi$; i.e.

$$b_\Delta(k) := \varphi(k - \Delta) \overset{z}{\leftrightarrow} B_\Delta(z) = \sum_{k \in Z} \varphi(k - \Delta) z^{-k}. \tag{9}$$

To implement this formula, we also need to evaluate the expansion coefficients of the function $s(x) \in V(\varphi)$ that interpolates $s[k]$. It can be shown that this interpolation problem has a unique solution if and only if $B_0(e^{j\omega}) = B_{\Delta=0}(e^{j\omega})$ satisfies the stability condition (4). Moreover, this solution can be obtained very simply by digital filtering

$$s[k] = s(x)|_{x=k} \Leftrightarrow c(k) = (b_0)^{-1} * s[k] \tag{10}$$

where the inverse filter is given by

$$(b_0)^{-1}(k) \overset{z}{\leftrightarrow} \frac{1}{\sum_{k \in Z} \varphi[k] z^{-k}}. \tag{11}$$

This result also provides the impulse response of the corresponding interpolator

$$\begin{aligned} \varphi_{\mathrm{int}}(x) &= \sum_{k \in Z} (b_0)^{-1}(k) \varphi(x - k) \overset{\text{Fourier}}{\longleftrightarrow} \Phi_{\mathrm{int}}(\omega) \\ &= \frac{\Phi(\omega)}{\sum_{k \in Z} \Phi(\omega + 2\pi k)} \end{aligned} \tag{12}$$

which is an important descriptor of the algorithm. Since (12) implies that $\Sigma_{k \in Z}\, \Phi_{\mathrm{int}}(\omega + 2\pi k) = 1$, the function $\varphi_{\mathrm{int}} \in V(\varphi)$ takes the value one at the origin and is zero for all other integers (interpolation property). Obviously, the prefiltering step in (10) can be avoided if φ already has this property (i.e., $\varphi = \varphi_{\mathrm{int}}$).

The whole computational procedure, which is equivalent to a single convolution with the translation filter $t_\Delta(k) := \varphi_{\mathrm{int}}(k - \Delta) = b_\Delta * (b_0)^{-1}(k)$, is summarized in Fig. 2(a). If φ is compactly supported, it is usually more

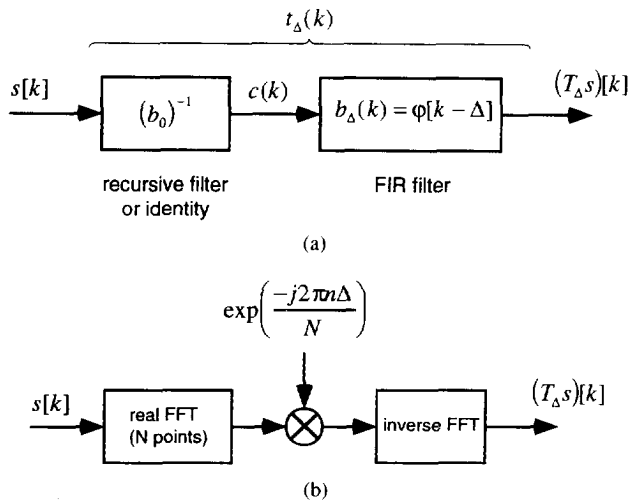


Fig. 2. Block diagram of a digital filter-based translation operator: (a) implementation in the signal domain; (b) implementation in the frequency domain (sinc-interpolator).

advantageous to evaluate the translation in the spatial domain. In this case, b_Δ is a finite impulse response (FIR) filter, and $(b_0)^{-1}$ is an all pole infinite impulse response (IIR) filter, which can usually be implemented using the fast recursive technique described in [16].

B. Order of Accuracy

The quality of an interpolator will depend on its ability to reproduce all polynomials up to a certain degree n. This maximum degree provides the order of the interpolator $N = n + 1$, which, in the convention of approximation theorists, is one more than the degree. It turns out that this property has a relatively simple interpretation in the frequency domain, a result that deserves to be better publicized for signal processing.

Proposition 1: An interpolator with generating function φ will reproduce all polynomials up to degree $N - 1$ if and only if $\Phi(\omega)$ is nonvanishing at the origin and has zeros of at least multiplicity N at all nonzero frequencies that are integer multiples of 2π.

This result was first described by Schoenberg in his landmark paper on splines [17]. It also plays a crucial role in the characterization of the approximation power of certain finite element methods [18] (Strang-Fix conditions), and more recently, in the theory of the wavelet transform [19]. We have chosen here to present our own version of the proof, which uses rather standard signal processing concepts and also provides some further insights into the interpolation process.

Specifically, the condition for an Nth order interpolation is that there exists a certain function $\varphi_{\text{int}}^N \in V(\varphi)$ (not necessarily unique) such that

$$\forall x \in R, \quad \sum_{k \in Z} k^n \varphi_{\text{int}}^N(x - k) = x^n, n = 0, \cdots, N - 1. \tag{13}$$

In other words, φ_{int}^N provides an exact interpolation formula for all polynomials of degree lesser or equal to $n = N - 1$. As shown in the Appendix, condition (13) can be rewritten in the equivalent form

$$\forall x \in R, \quad \sum_{k \in Z} \varphi_{\text{int}}^N(x - k) = 1 \tag{14}$$

$$\forall x \in R, \quad \sum_{k \in Z} (x - k)^n \varphi_{\text{int}}^N(x - k) = 0, \quad n = 1, \cdots, N - 1. \tag{15}$$

The key to the proof is then to identify these sums as the periodization of the functions $x^n \varphi_{\text{int}}^N(x)$ and recall that this operation corresponds to a sampling in the frequency domain. Taking the Fourier series of (14) and (15), we get the equivalent frequency domain interpretation of these constraints

$$\Phi_{\text{int}}^N(\omega)|_{\omega = 2\pi k} = \delta[k], \tag{16}$$

$$\left.\frac{\partial^n \Phi_{\text{int}}^N(\omega)}{\partial \omega^n}\right|_{\omega = 2\pi k} = 0, \qquad n = 1, \cdots, N - 1 \tag{17}$$

where $\delta(k)$ denotes the discrete unit impulse. The condition in Proposition 1 then follows directly from the fact that $\Phi_{\text{int}}^N(\omega) = C(e^{j\omega})\Phi(\omega)$ where $C(e^{j\omega})$ is 2π-periodic, since $\varphi_{\text{int}}^N \in V(\varphi)$ by definition. Conversely, if $\Phi(\omega)$ has the required properties, it is always possible to construct a function $\varphi_{\text{int}}^N(x) = \Sigma c(k)\varphi(x - k)$ that also satisfies the additional $(N + 1)$ linear constraints for $\omega = 0$. An example of such a function is the interpolator defined by (12), but there are also many other ones since (13) is only a *quasi*-interpolation requirement (i.e., up to order N).

For $N = 1$, the required condition is (14). The shortest possible function that satisfies this constraint for all x is obviously $\text{rect}(x)$. The most direct way of obtaining a function that satisfies the condition in Proposition 1 is then to convolve $\text{rect}(x)$ with itself an appropriate number of times. This simple construction scheme yields the B-spline of degree n. This is optimal in the sense that it has the shortest support for the given order of accuracy.

With an Nth order interpolator, we have at least the insurance that the signal models prior and after translation are in agreement over the N first terms of their Taylor series. Hence, it is possible to recover the original sequence $s[k]$ from its translated version $(T_\Delta s)[k]$ within an $O(h^N)$ precision, where the step $h \leq \frac{1}{2}$ corresponds to the distance of Δ to the nearest integer.

C. Specific Interpolation Models

Let us now consider some examples.

- *Piecewise linear model* $(N = 2)$: In this case, the generating function φ is the tent function (or B-spline of degree 1)

$$\beta^1(x) = \begin{cases} 1 - |x|, & |x| < 1 \\ 0, & \text{otherwise,} \end{cases} \tag{18}$$

and no prefiltering is necessary. The corresponding first order translation filter $t_\Delta^1(k) = \beta^1(k - \Delta)$ can be implemented with two multiplications and one addition per input sample.

TABLE I
COMPARISON OF THE VARIOUS ALGORITHMS IN TERMS OF RMS ERROR AFTER 16 SUCCESSIVE ROTATIONS OF 22.5° EACH (FULL CIRCLE ROTATION)

Algorithm	Murray	Lenna	Circles	CPU time (16 rotations)
Non-separable:				
INT-0 (near. neighbor)	17.4372	21.1785	70.9194	3s
INT-1 (bilinear)	14.2957	19.3504	69.9067	5s
INT-Keys	7.34571	12.8394	56.8026	42s
INT-3 (cubic spline)	5.26271	9.74491	34.6474	43s
Three-pass separable:				
SEP3-1 (=*ls*0)	16.6941	21.0507	70.5951	5s
SEP3-3 (=*ls*1)	5.85241	10.437	42.3718	7s
SEP3-5	4.48279	8.13631	23.0364	11s
SEP3-7 (=*ls*3)	3.91884	7.12958	15.0174	14s
SEP3-sinc	4.42176	6.20883	4.15621	31s
Two-pass separable:				
SEP2-*ls*0	14.791	19.7454	70.1329	11s
SEP2-*ls*1	5.55867	10.2907	38.7047	21s
SEP2-*ls*3	4.33042	8.20193	20.7148	61s

- *Keys' cubic interpolant* ($N = 3$): The cubic interpolation function proposed by Keys is given by (cf. [4])

$$\varphi_a^3(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1, & 0 \le |x| < 1 \\ a(|x|^3 - 5|x|^2 + 8|x| - 4), & 1 \le |x| < 2 \\ 0, & 2 \le |x|. \end{cases} \tag{19}$$

Although this function is piecewise cubic, the order of the corresponding interpolator is only $N = 3$ for the optimal choice $a = -\frac{1}{2}$, and less otherwise. The good news is that no prefiltering is necessary.

- *Cubic spline model* ($N = 4$): The generating function for cubic spline interpolants is the cubic B-spline which is given by

$$\beta^3(x) = \begin{cases} 2/3 - |x|^2 + |x|^3/2, & 0 \le |x| < 1 \\ (2 - |x|)^3/6, & 1 \le |x| < 2 \\ 0, & 2 \le |x|. \end{cases} \tag{20}$$

This formula can be used to evaluate the postfilter $b_\Delta^3(k) = \beta^3(k - \Delta)$, which has at most four nonzero coefficients. The corresponding inverse filter is

$$(b_0^3)^{-1}(k) \overset{z}{\leftrightarrow} \frac{6}{z + 4 + z^{-1}}. \tag{21}$$

By decomposing it into a cascade of first-order causal and anticausal exponentials, it can be implemented recursively with two multiplications and two additions per input sample [6].

- *Polynomial splines of degree* n ($N = n + 1$): We can also construct higher-order polynomial spline interpolants by choosing $\varphi(x) = \beta^n(x)$, where β^n is Schoenberg's central B-spline of degree n [17]. This function is usually constructed from the $(n+1)$-fold convolution of $\mathrm{rect}\,(x)$. An equivalent definition that is more useful for the evaluation of $b_\Delta^n(k) = \beta^n[k - \Delta]$ is

$$\beta^n(x) = \sum_{k=0}^{n+1} \frac{(-1)}{n!} \binom{n+1}{k} \left(x + \frac{n+1}{2} - k\right)_+^n \tag{22}$$

where $(x)_+ = \max(0, x)$. In this case, b_Δ^n has at most $(n + 1)$ nonzero coefficients. The most efficient way to implement the corresponding inverse filter $(b_0^n)^{-1}$ is to use the recursive algorithm described in [16]; that reference also contains explicit filter formulas for $n = 0, \cdots 7$ (cf. Table I). With this technique, an nth degree translation (n odd) can be implemented with as few as $2n$ multiplications and $2n - 1$ additions per sample point.

- *Bandlimited model* ($N \to \infty$): For $\varphi(x) = \mathrm{sinc}\,(x)$, $V(\varphi)$ corresponds to the class of bandlimited functions considered in Shannon's sampling theorem [20]. It is worth noting that these interpolants can also be interpreted as splines of infinite order [21]. In this case, $(b_0)^{-1}$ is the identity and the translation filter is $t_\Delta^\infty(k) = \mathrm{sinc}\,(k - \Delta)$. To evaluate its transfer function, we start by writing the Fourier transform of the continuous-time function $\mathrm{sinc}\,(x - \Delta)$: $\Phi_\Delta^\infty(\omega) = e^{-j\omega\Delta}\,\mathrm{rect}\,(\omega/2\pi)$. We then express the effect of sampling in the Fourier domain, which yields

$$T_\Delta^\infty(e^{j\omega}) = \sum_{k \in Z} \Phi_\Delta^\infty(\omega + 2\pi k) = e^{-j\omega\Delta}. \tag{23}$$

Note that this result is consistent with the shift property of the Fourier transform. Because of the slow decay of the interpolation kernel, the filter is most effectively implemented in the Fourier domain, as illustrated by the block diagram in Fig. 2(b).

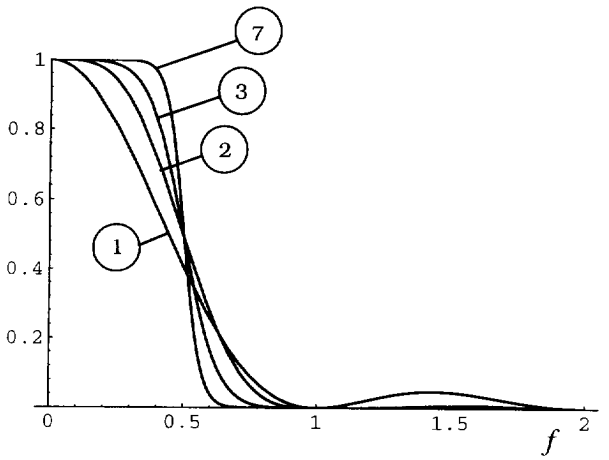


Fig. 3. Frequency responses of various interpolators with increasing orders of accuracy. (1) Piecewise linear; (2) small kernel cubic; (3) cubic spline; (7) spline of order 7.

The frequency responses of these various interpolators are represented in Fig. 3. Note how the flatness of the response at the origin and the attenuation in the higher-frequency band improve with the order N. The reason for the superiority of cubic spline interpolation over the small kernel cubic convolution method is also quite apparent from this graph.

III. LEAST-SQUARES DESIGN OF THE TRANSLATION OPERATOR

The translation method that we describe next uses the same interpolation models as before, but, instead of simply resampling, it computes the orthogonal projection of the translated function $T_\Delta s(x)$ in $V(\varphi)$. For the given interpolation model, this is the optimal way to discretize the translation operator

because it minimizes the L_2-error between the exact translation and its approximation in $V(\varphi)$ (least-squares solution). The resulting approximation is then represented by its sample values, which provide a full characterization (cf. (10)). We will see that this approach performs better than the previous resampling scheme, which was not explicitly designed to minimize the error.

Specifically, let P_V denote the orthogonal projection operator onto $V(\varphi)$. The least-squares approximation of $T_\Delta s(x)$ in $V(\varphi)$ can be expressed as (cf. [15])

$$(P_V T_\Delta s)(x) = \sum_{k \in Z} c_\Delta(k) \varphi(x - k) \tag{24}$$

where the c_Δ's are obtained from the L_2-inner product between the function to be approximated and corresponding *dual* analysis function

$$c_\Delta(k) = \langle T_\Delta s(x), \mathring{\varphi}(x - k) \rangle. \tag{25}$$

The dual generating function $\mathring{\varphi}(x) \in V(\varphi)$ is unique in the sense that it satisfies the biorthogonality condition

$$\langle \varphi(x - k), \mathring{\varphi}(x - l) \rangle = \delta[k - l]. \tag{26}$$

Next, we introduce the auxiliary function

$$\varphi_{ls}(x) = \langle \varphi(u), \mathring{\varphi}(u - x) \rangle = (\varphi^T * \mathring{\varphi})(x) \tag{27}$$

where $\varphi^T(x) = \varphi(-x)$. We then use the formula $T_\Delta s(x) = \Sigma_{l \in Z}\, c(l) \varphi(x - \Delta - l)$, and rewrite (25) as

$$c_\Delta(k) = \sum_{l \in Z} c(l) \langle \varphi(x - \Delta - l), \mathring{\varphi}(x - k) \rangle \tag{28}$$

or, equivalently

$$c_\Delta(k) = \sum_{l \in Z} c(l) \varphi_{ls}(k - \Delta - l) = t_\Delta^{ls} * c(k) \tag{29}$$

where $t_\Delta^{ls}(k) = \varphi_{ls}(k - \Delta)$. Finally, by combining this result with (10) and discretizing (24) at the integers, we get

$$\begin{aligned}(P_V T_\Delta s)[k] &= b_0 * c_\Delta(k) = b_0 * t_\Delta^{ls} * (b_0)^{-1} * s[k] \\ &= t_\Delta^{ls} * s[k]\end{aligned} \tag{30}$$

where the term $b_0(k)$ represents the sampled version of the basis function in (24). What remains after simplification is the convolution of $s[k]$ with the digital filter t_Δ^{ls}. Further, this result can also be interpreted as an interpolation formula

$$(P_V T_\Delta s)[k] = \sum_{l \in Z} s[l] \varphi_{ls}(x - l)|_{x = k - \Delta} \tag{31}$$

where we note that $\varphi_{ls}(x)$ is a true interpolation function (i.e., $\varphi_{ls}[k] = \delta[k]$); a property that follows directly from (26)).

What this derivation effectively shows is that the least-squares solution for the translation in $V(\varphi)$ is equivalent to an interpolation in the space $V(\varphi_{ls}) = V(\varphi^T * \mathring{\varphi}) = V(\varphi^T * \varphi)$, and that it can therefore also be implemented using the general procedure described in Section II-A. In general, the least-squares solution will result in a higher-order interpolation. In particular, if φ is a B-spline of degree n then $\varphi_{ls}(x)$ is the cardinal (or fundamental) spline of degree $2n + 1$ [22]; the least-squares solution therefore corresponds to a polynomial spline interpolation of degree $2n + 1$ (i.e., twice the order).

It is clear that the interpolation and least-squares solutions are identical for the bandlimited case because $\mathrm{sinc} * \mathrm{sinc}\,(x) = \mathrm{sinc}\,(x)$. Moreover, the translation operation with this particular model is error-free because $T_\Delta s(x)$ remains in the original approximation space (shift-invariance property). Other related aspects of the optimality of bandlimited representations are discussed in [23].

In general, the least-squares solution for a particular space $V(\varphi)$ will get us one step closer to the optimal sinc solution. Specifically, if we start with $\varphi_0(x) = \varphi(x)$ and iterate the least-squares solution over and over again then the resulting interpolating function $\varphi_i(x) = \varphi_{i-1}^T * \mathring{\varphi}_{i-1}(x)$ will converge to $\mathrm{sinc}\,(x)$ as $n = 2^i$ goes to infinity. This general convergence result comes as a corollary of Theorem 9 in [15]; it holds for any starting function φ_0 under two relatively mild conditions: (1) $\Phi_0(\omega) = O(|\omega|^{-r})$ for some $r > \frac{1}{2}$, and (2) $\forall \omega \in [-\pi, \pi], \Phi_0(\omega) \geq \Phi_0(\pi)$.

IV. Results and Discussion

A. Experimental Setting

The various interpolation models in Section II-C were compared in their efficiency for implementing the three-pass image rotation algorithm (SEP3) described in the introduction. All computations were performed in floating point, and the intermediate results were stored in real format to minimize the effect of round-off errors. The final rotated images were all truncated to the nearest integer. In all cases, the one-dimensional translation filtering was performed in place, using periodic boundary conditions. All spline-based translations for SEP3-n, $n \in \{1, 3, 5, 7\}$, were evaluated in the spatial domain according to the procedure outlined in Fig. 2(a). The sinc-based translation for SEP3-∞ = SEP3-sinc was implemented in the Fourier domain using a special version of the FFT algorithm for real signals (cf. Fig. 2(b)).

For comparison, we also implemented our own version of the two-pass rotation algorithm (SEP2) using various spline models. The difficulty with this approach is that the first pass involves a spatial signal contraction by a factor $1/\cos\theta$, which may introduce aliasing artifacts if it is not done properly. To minimize these errors, we performed the various 1-D geometrical transformations using the least-squares rescaling algorithm described in [24]. We found this approach to be superior to a simple interpolation. In fact, the improvement is at least as good as that obtained by switching from an nth degree spline interpolation to a $2n + 1$ degree model in the simple translation case (cf. Section III). The versions of the two-pass algorithm that were implemented are the least-squares solutions for piecewise constant, linear, and cubic spline signal models (SEP2-lsn, $n \in \{0, 1, 3\}$).

Several standard nonseparable rotation algorithms (INT-n) were also included in the comparison. The corresponding 2-D interpolation formulas for computing the value of the image at location (x, y) are all of the form

$$s(x,y) = \sum_{k=k_0}^{k_0+L-1} \sum_{l=l_0}^{l_0+L-1} c(k,l)\varphi(x-k)\varphi(y-l) \qquad (32)$$

where k_0 and l_0 are the nearest integer truncation of $(x - L/2)$ and $(y - L/2)$, respectively. The explicit equations for φ for the bilinear (INT-1 with $L = 2$), short kernel cubic (INT-keys with $L = 4$), and cubic spline models (INT-3 with $L = 4$) are (18), (19) with $a = -1/2$, and (20), respectively. Note that such a nonseparable interpolation requires at least L^2 multiplications and L^2 additions per pixel, plus the cost of the evaluation of values of the basis functions themselves, which may in fact be the most demanding task, especially for higher-order models (e.g., $n + 1$ multiplications and n additions for evaluating a polynomial of degree n). In the case of the cubic spline, the procedure also included a two-dimensional prefiltering step to pre-compute the cubic B-spline coefficients of the image [6].

In our experiments, we used three 256 × 256 test images: Murray (Fig. 6), who is an occasional collaborator, Lena (who is not), and circles (Fig. 7). The concentric circle pattern was generated by sampling a radial chirp-like cosine function with a period that increases linearly from the center ($T_{\min} = 2$) to the periphery ($T_{\max} = 4$). Although this pattern is circularly symmetric, it poses a challenge for most rotation algorithms because it contains many high frequencies.

B. Results

First, we conducted a series of back and forth rotations at various angles and computed the residual root mean square error on the 128 × 128 central portion of the image to factor out boundary effects. The results of these experiments are displayed in the graphs in Fig. 4. The results for Lena (which are not included) were qualitatively very similar to those for Murray. As expected, the performance of all algorithms improves as a function of the degree n. Near-perfect rotation is achieved by the three pass FFT algorithm (SEP3-sinc); in this case, the residual error is entirely due to roundoff effects (integer truncation). SEP3-7 is also very competitive and essentially outperforms all other procedures. For a given degree n, the nonseparable algorithm (INT) tends to perform better than the corresponding separable algorithms (SEP2 and SEP3), although this advantage is not as significant for higher orders ($n = 3$). Also note that the two-pass algorithm progressively loses its efficiency with increasing angles, whereas the performance of both nonseparable (INT-n) and three-pass separable (SEP3-n) algorithms is hardly angle-dependent at all. This serious flaw of SEP2 can be explained by the fact that the transformation is unable to retain all high-frequency information because of the signal compression by $\cos\theta$ that occurs during the first pass.

Although the back and forth rotation experiment illustrates our point, it does not necessarily provide a good indication of the quality of the rotated image itself. For instance, the INT-0 algorithm, which performs a mere data shuffling, is perfectly reversible with an RMS error of zero. At the other extreme, the SEP3-sinc algorithm also turns out to be perfectly reversible because the sinc-translation is lossless and the three-pass factorization (2) is symmetrical. In other words, the sequence of transformations for a rotation by $-\theta$ is exactly the inverse of that for a rotation by θ. This property has been verified experimentally with images stored in floating point format.

To get a better idea of the effect of a single rotation, we chose to concentrate on the circularly symmetric circle pattern, which has the advantage of providing its own rotated reference. Fig. 5 provides a visual comparison of the performance of the

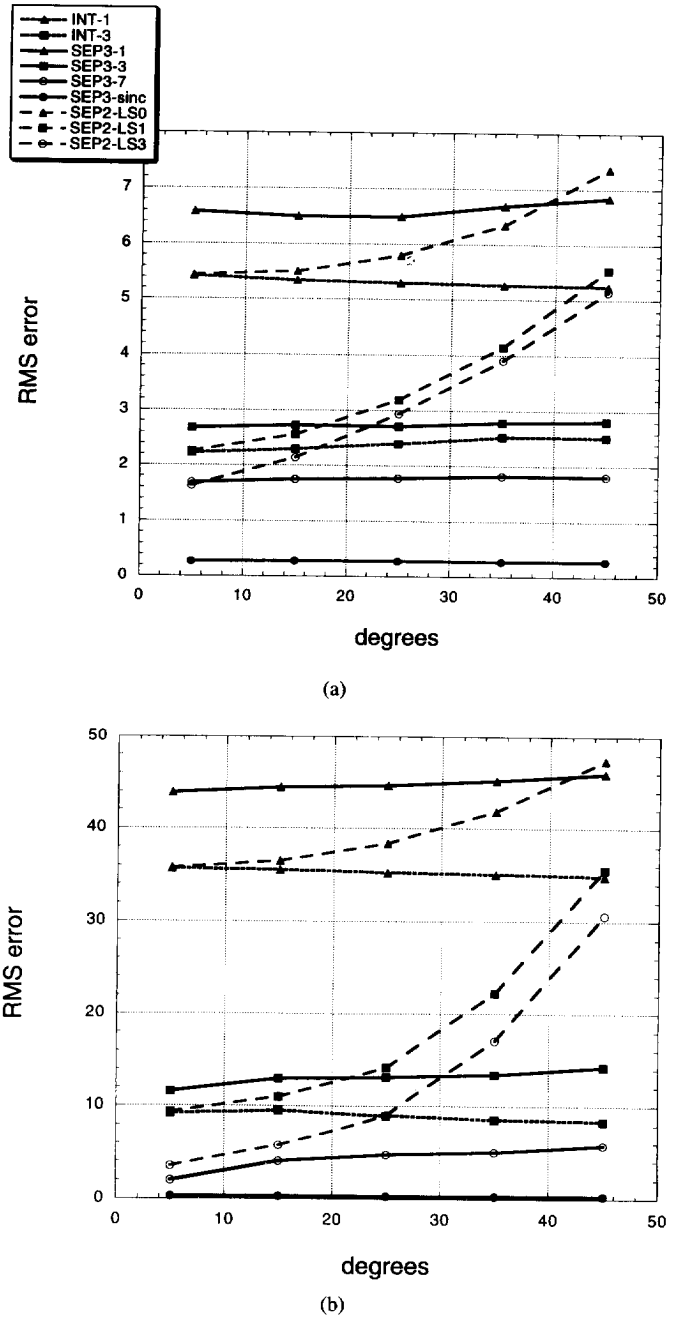


Fig. 4. Residual error as a function of the angle θ for various rotation algorithms: (a) for the image Murray shown in Fig. 6; (b) for the circle pattern shown in Fig. 7.

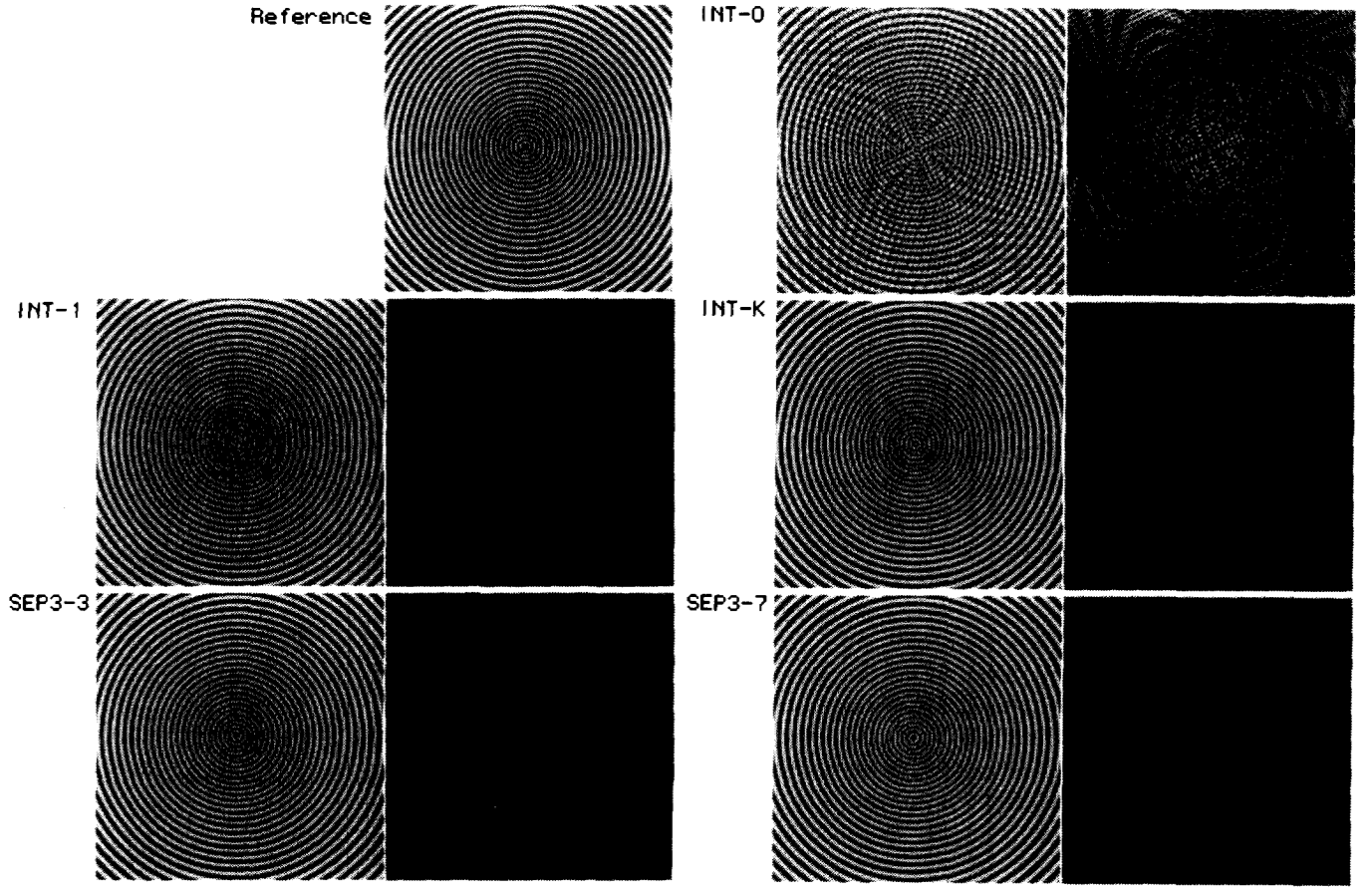


Fig. 5. Examples of rotations by 37° of a circular symmetric pattern and the corresponding absolute value of the error amplified by a factor of two. The reference 128 × 128 region of interest is shown on the upper left corner. The corresponding RMS error values are: $\varepsilon = 40.09$ for INT-0, $\varepsilon = 23.66$ for INT-1, $\varepsilon = 13.40$ for INT-Keys, $\varepsilon = 9.24$ for SEP3-3, and $\varepsilon = 4.31$ for SEP3-7.

various algorithms in the case of a single rotation; it displays both the central portion of resulting images rotated by 37° and the absolute value of the difference with the original circle pattern. This example also provides a good illustration of the type of artifacts that may result from the use of the standard (low accuracy) INT-0 and INT-1 methods.

To obtain an even more challenging benchmark, we considered the cumulative effect of 16 successive rotations of $\pi/8$ each, at the end of which the image is back in its initial position. Some examples of results are shown in Figs. 6 and 7. Table I provides a detailed performance comparison of the various algorithms, including the total CPU time on a Silicon Graphics SGX workstation. The results are quite dramatic and emphasize the type of artifacts (blocking and smoothing) introduced by lower-order methods. The circle pattern also permits a direct visualization of the loss of high-frequency information that typically occurs in the center of the image (cf. Fig. 7). This series of experiments clearly demonstrates—both qualitatively and quantitatively—the superiority of the high-accuracy methods (SEP3-$n, n \geq 3$, and SEP3-sinc). Note that the small cubic kernel method (INT-Keys), which is usually considered one of the best available methods [3], does not score particularly well by comparison. It has one of the highest computational costs and is outperformed by most of the newer methods presented here. Even in its own category, it is more advantageous to use the nonseparable cubic spline interpolation (INT-3), which has approximately the same complexity but has one more order of accuracy (cf. Section II.B). This result clearly shows the relevance of the optimality property of B-splines (shortest support for a given accuracy); this feature is especially important in the nonseparable case where the cost of the evaluation of the B-spline coefficients (prefiltering) is almost negligible. The SEP3 algorithms are all extremely competitive in terms of their computational efficiency. At the lower end, SEP3-3, which is nearly as efficient as INT-3, is almost as fast as bilinear interpolation (INT-1). In summary, the three-pass procedure does usually provide better-quality pictures with fewer computations.

V. Conclusion

In this paper, we have presented a general formulation and analysis of convolution-based interpolation techniques. We have applied those results to the design of high-quality three-pass rotation algorithms. We have also presented a detailed evaluation of these algorithms and a comparison with the standard methods. The advantageous features of the present approach as applied to this particular task are as follows:

- *Simplicity*: Image rotations can be performed in an entirely separable fashion using one-dimensional convo-

Fig. 6. Example of results after 16 successive rotations of 22.5° each. The displayed region of interest is enlarged by a factor of two using simple pixel replication. (a) Original image (Murray) with the 128×128 region of interest; (b) nonseparable nearest neighbor rotation (INT-0); (c) nonseparable bilinear rotation (INT-1); (d) nonseparable short cubic rotation (INT-Keys); (e) three-pass seventh order spline rotation (SEP3-7); (f) three-pass sinc rotation (SEP3-sinc).

lutions only; this property simplifies both the design and the implementation. Separability turns out to be a significant advantage for hardware implementation and parallelization. It also facilitates the processing of very large images since the memory requirements can be limited to one column or one row at a time.

- *Efficiency and flexibility*: The approach is extremely fast and can achieve any desired degree of accuracy by increasing the order of the interpolator. For polynomial spline models up to degree 7, the convolutions are most efficiently computed in the spatial domain. The highest-quality image rotations are obtained with the sinc interpolator, which can be implemented using FFT's. Even in this limiting case, the algorithm is substantially faster than most nonseparable small kernel interpolation methods.
- *Performance improvement*: The approach is especially advantageous for performing high-quality image rotations using higher-order interpolation models ($n \geq 3$). Its complexity increases linearly with the degree $O(n)$, while the conventional nonseparable interpolation approach requires $O(n^2)$ to $O(n^3)$ operations per pixel, depending on whether or not one includes the cost of the evaluation of the basis functions. Another important factor is that the difference in quality between the two approaches becomes less and less significant for higher-order models. In other words, we get the best computational improvement for the highest-quality results.

APPENDIX

In this Appendix, we formally prove the equivalence between (13) and (14)–(15). For this purpose, we consider the term

$$\begin{aligned}
&\sum_{k\in Z}(x-k)^n\varphi_{\text{int}}^N(x-k)\\
&\quad=\sum_{k\in Z}\sum_{j=0}^{n}\binom{n}{j}x^j(-k)^{n-j}\varphi_{\text{int}}^N(x-k)\\
&\quad=\sum_{j=0}^{n}\binom{n}{j}x^j(-1)^{n-j}\\
&\qquad\cdot\left(x^{n-j}-x^{n-j}+\sum_{k\in Z}k^{n-j}\varphi_{\text{int}}^N(x-k)\right)\\
&\quad=\sum_{j=0}^{n}\binom{n}{j}x^j(-x)^{n-j}+\sum_{j=0}^{n}\binom{n}{j}x^j(-1)^{n-j}\\
&\qquad\cdot\left(\sum_{k\in Z}k^{n-j}\varphi_{\text{int}}^N(x-k)-x^{n-j}\right).
\end{aligned}$$

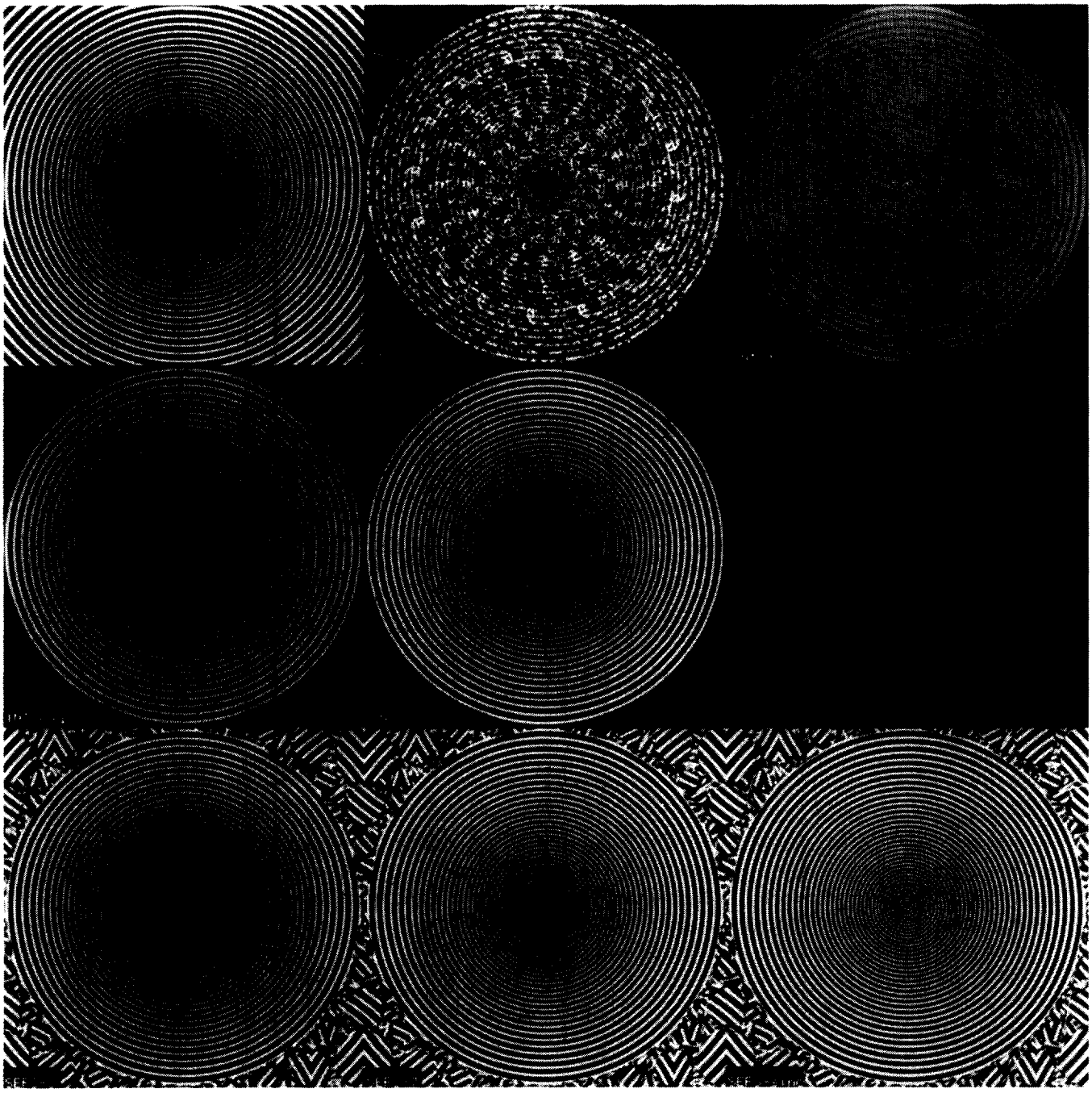


Fig. 7. Examples of image displays after 16 successive rotations of 22.5° each. From upper left to bottom right: (a) original 256 × 256 circle pattern; (b) nonseparable nearest neighbor rotation (INT-0); (c) nonseparable bilinear rotation (INT-1); (d) nonseparable short cubic rotation (INT-Keys); (e) nonseparable cubic spline rotation (INT-3); (f) three-pass cubic spline rotation (SEP3-3); (g) three-pass seventh-order spline rotation (SEP3-7); (h) three-pass sinc rotation (SEP3-sinc).

Since the first term is simply the development of $(x - x)^n$, we end up with the relation

$$\sum_{k \in Z} (x-k)^n \varphi_{\text{int}}^N(x-k) = \sum_{j=0}^{n} \binom{n}{j} x^j (-1)^{n-j} \cdot \left(\sum_{k \in Z} k^{n-j} \varphi_{\text{int}}^N(x-k) - x^{n-j} \right). \quad \text{(A1)}$$

This last identity is then used to establish the equivalence between (13) and (14)–(15) by working our way up from $n = 0$ to $N - 1$.

REFERENCES

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

[2] S. K. Park and R. A. Showengerdt, "Image reconstruction by parametric convolution," *Comput. Vision, Graphics, Image Processing,* vol. 20, no. 3, pp. 258–272, Sept. 1983.

[3] J. A. Parker, R. V. Kenyon, and D. E. Troxel, "Comparison of interpolating methods for image resampling," *IEEE Trans. Med. Imaging,* vol. MI-2, pp. 31–39, 1983.

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

[5] 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.

[6] 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, Mar. 1991.

[7] E. Catmul and A. R. Smith, "3-D transformation of images in scan-line order," in *SIGGRAPH Proc.,* vol. 14, 1980, pp. 279–285.

[8] D. Fraser, "Comparison of high spatial frequencies of two-pass and one-pass geometric transformation algorithms," *Comput. Vision, Graphics, Image Processing,* vol. 46, pp. 267–283, 1989.

[9] D. E. Friedman, "Two-dimensional resampling of line scan imagery by one-dimensional processing," *Photogrammetric Eng. Remote Sensing,* vol. 47, pp. 1459–1467, Oct. 1981.

[10] N. Tsuchida, Y. Yamada, and M. Ueda, "Hardware for image rotation by twice skew transformation," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-35, pp. 527–531, Apr. 1987.

[11] D. Fraser, R. A. Schowengerdt, and I. Briggs, "Rectification of multichannel images in mass storage using image transposition," *Comput. Vision, Graphics, Image Processing,* vol. 29, pp. 23–36, 1985.

[12] H. Kiesewetter and A. Graf, "Rotation in digital grids and corresponding models," Zentral Institut für Kybernetik und Informations Prozesse, Akademie der Wissenschaften der DDR Tech. Rep., 1985.

[13] A. W. Paeth, "A fast algorithm for general raster rotation," in *Proc. Graphics Interface '86—Vision Interface '86,* 1986, pp. 77–81.

[14] P. E. Danielsson and M. Hammerin, "High-accuracy rotation of images," *CVGIP: Graphical Models and Image Processing,* vol. 54, no. 4, pp. 340–344, July 1992.

[15] A. Aldroubi and M. Unser, "Sampling procedures in function spaces and asymptotic equivalence with Shannon's sampling theory," *Numer. Funct. Anal. Optimiz.,* vol. 15, nos. 1 and 2, pp. 1–21, Feb. 1994.

[16] 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.

[17] I. J. Schoenberg, "Contribution to the problem of approximation of equidistant data by analytic functions," *Q. Appl. Math.,* vol. 4, pp. 45–99 and pp. 112–141, 1946.

[18] 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.

[19] G. Strang, "Wavelets and dilation equations: A brief introduction," *SIAM Rev.,* vol. 31, pp. 614–627, 1989.

[20] C. E. Shannon, "Communication in the presence of noise," *Proc. IRE,* vol. 37, pp. 10–21, Jan. 1949.

[21] A. Aldroubi, M. Unser, and M. Eden, "Cardinal spline filters: Stability and convergence to the ideal sinc interpolator," *Signal Processing,* vol. 28, no. 2, pp. 127–138, Aug. 1992.

[22] M. Unser, A. Aldroubi, and M. Eden, "Polynomial spline signal approximations: Filter design and asymptotic equivalence with Shannon's sampling theorem," *IEEE Trans. Inform. Theory,* vol. 38, pp. 95–103, Jan. 1992.

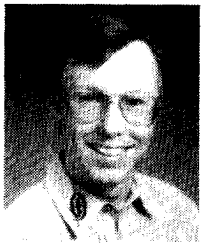
[23] M. Unser, "On the optimality of ideal filters for pyramid and wavelet signal approximation," *IEEE Trans. Signal Processing,* vol. 41, pp. 3591–3596, Dec. 1993.

[24] 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, Mar. 1995.

Michael Unser (M'88–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 of Lausanne, Switzerland.

He is currently a Visiting Scientist with the Biomedical Engineering and Instrumentation Program, National Institutes of Health, Bethesda, MD, which he joined in 1985. He has also been affiliated with INSERM (French National Institutes of Health and Biomedical Research) since April 1988. His research interests include the application of image processing and pattern recognition techniques to various biomedical problems, multiresolution algorithms, wavelet transforms, and the use of splines in signal processing.

Dr. Unser was awarded the Brown–Bowery Prize in 1984 for his work on texture analysis and automated inspection. He is the author of more than 50 published journal papers. He serves as an associate editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE SIGNAL PROCESSING LETTERS, and is a member of the Multidimensional Signal Processing Committee of the IEEE Signal Processing Society. He is also on the editorial board of *Signal Processing* and *Pattern Recognition.*



Philippe Thévenaz (PM'95) was born in Lausanne, Switzerland. He graduated from the Lausanne Swiss Federal Institute of Technology (EPFL) in January 1986, with a diploma in microtechnology. He received the Ph.D. degree in June 1993, with a thesis on the use of the linear prediction residue for text-independent speaker recognition.

He joined the Institute of Microtechnology of the University of Neuchâtel (IMT), Switzerland, in 1986, where he worked as Scientific Collaborator and Assistant, first in the domain of image processing (optical flow), then in the domain of speech processing (speaker recognition). He is currently a Visiting Fellow with the Biomedical Engineering and Instrumentation Program, National Institutes of Health (NIH), Bethesda, MD. His research interests include splines and multiresolution signal representations, geometric image transformations, and biomedical image registration.

Leonid Yaroslavsky was born in Kharkov, Ukraine, in 1940. He received the Diplom-Engineer degree (cum laude) in 1961 in radio engineering (M.S. equivalent) from the Kharkov Polytechnic Institute, the Candidate of Sciences degree in radio engineering (Ph.D. equivalent) from the Moscow Institute of Energy in 1968, and the Doctor of Sciences in Physics and Mathematics degree in optics from the State Optical Institute, St. Petersburg, Russia, in 1982.

Since 1962, he has been associated with the Institute of Information Transmission Problems, Russian Academy of Sciences. In 1991 he was a Guest Professor with the Department of Applied Optics, University of Erlangen, Germany. At this time, he is a Visiting Scientist with the Biomedical Engineering and Instrumentation Program at the National Institutes of Health, Bethesda, MD. His main fields of interest include multidimensional digital signal and image processing, specifically image filtering, enhancement, restoration, segmentation, and analysis, image representation, interpolation and spatial transformations, multicomponent image processing, speckle imaging and computer holography, image scanning, sampling, quantization and visualization, image coding. He is the author and co-author of several books and chapters in several books on these specialties.

Dr. Yaroslavsky is a member of the Optical Society of America.