Comparison of Interpolating Methods for Image Resampling

J. ANTHONY PARKER, ROBERT V. KENYON, AND DONALD E. TROXEL

***Abstract*—When resampling an image to a new set of coordinates (for example, when rotating an image), there is often a noticeable loss in image quality. To preserve image quality, the interpolating function used for the resampling should be an ideal low-pass filter. To determine which limited extent convolving functions would provide the best interpolation, five functions were compared: A) nearest neighbor, B) linear, C) cubic B-spline, D) high-resolution cubic spline with edge enhancement ($a = -1$), and E) high-resolution cubic spline ($a = -0.5$). The functions which extend over four picture elements (C, D, E) were shown to have a better frequency response than those which extend over one (A) or two (B) pixels. The nearest neighbor function shifted the image up to one-half a pixel. Linear and cubic B-spline interpolation tended to smooth the image. The best response was obtained with the high-resolution cubic spline functions. The location of the resampled points with respect to the initial coordinate system has a dramatic effect on the response of the sampled interpolating function—the data are exactly reproduced when the points are aligned, and the response has the most smoothing when the resampled points are equidistant from the original coordinate points. Thus, at the expense of some increase in computing time, image quality can be improved by resampled using the high-resolution cubic spline function as compared to the nearest neighbor, linear, or cubic B-spline functions.**

Introduction

RESAMPLING is used for several different purposes in image processing. An image may be resampled to a finer matrix in order to improve its appearance for image display—reduce artifacts due to the boundary between picture elements (pixels). When an image is rotated by angles which are not a multiple of 90 degrees, resampling is required since the new coordinate points will not line up with the old points. In remotely sensed data, there are often distortions due to perspective or due to the atmosphere, which can be removed by resampling the image to an undistorted coordinate system [1], [2]. When registering images taken with different sensors or at different times, it may be necessary to resample the images so that the registration is accurate to subpixel locations.

Several of these uses of image resampling are valuable for medical imaging. But, resampling for image registration and image rotation are of particular interest for digital radioloy. Much of the early interest in digital radiology has been due to the ability to enhance images obtained after injection of contrast by subtraction of a mask image(s) taken before the injection [3]. There may be complex distortion between the image and the subtraction mask due to patient motion [4]. Since subtraction depends upon careful registration of the image and the subtraction mask, improved resampling algorithms may improve the accuracy of the registration and thus improve the quality of the images produced by digital subtraction angiography.

Several interpolating functions have been used for image resampling. The simplest is the nearest neighbor function, where the value of the new point is taken as the value of the old coordinate point which is located the nearest to the new point. Another algorithm frequently used is linear interpolation, where the new point is interpolated linearly between the old points. The next most complex functions use the four nearest points (two points in each direction). Cubic B-spline interpolating functions, which were investigated by Hou and Andrews [5], are positive everywhere and tend to smooth the resampled image. Cubic splines which are negative in the interval (1, 2) tend to preserve the original image resolution [6]-[9].

The choice of an interpolating function to be used for resampling depends upon the task being performed. An understanding of the frequency domain response of the interpolating functions may help in the choice of an interpolating function [9], [10], especially when the resampling is used prior to further image processing. This paper describes the resampling process and then examines the frequency domain response of five interpolating functions: nearest neighbor, linear, cubic B-spline, and two high-resolution cubic spline functions.

Resampling

Resampling is the process of transforming a discrete image which is defined at one set of coordinate locations to a new set of coordinate points. Resampling can be divided conceptually into two processes: interpolation of the discrete image to a continuous image and then sampling the interpolated image (Fig. 1). Frequently, resampling is used to increase the number of points in an image to improve its appearance for display. This process of filling in points between the data

Manuscript received May 25, 1983. This work was supported by the James Picker Foundation under Grant SA 76-6; by NASA Ames Research Center under Grant NAG 2-88; and by Beth Israel Hospital Core Laboratory under Grant RR-01302 from the General Clinical Research Centers Program of the Division of Research Sources, National Institutes of Health. Dr. Parker is the recipient of a Research Career Development Award, 5 K04 HL00465, from the National Heart, Lung, and Blood Institute.

J. A. Parker is with the Charles A. Dana Research Institute and the Joint Program in Nuclear Medicine, Departments of Radiology of the Beth Israel Hospital and Harvard Medical School, Boston, MA, 02215, and also with the Man-Vehicle Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139.

R. V. Kenyon and D. E. Troxel are with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139.

0278-0062/83/0300-0031$01.00

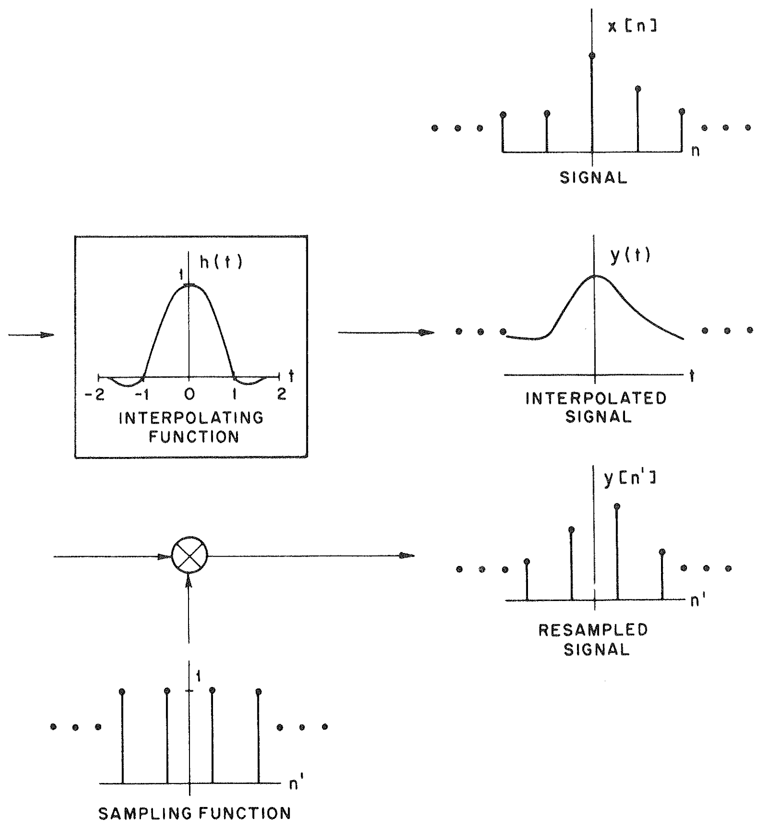


Fig. 1. Resampling. The process of resampling can be divided into two processes—interpolation followed by sampling; the effects of each of these operations can then be considered separately. A discrete function, $x[n]$, is convolved with an interpolating function, $h(t)$, to produce a continuous function, $y(t)$. The continuous function is then multiplied by a sampling function to produce a discrete function resampled at a new set of points, $y(n')$.

points is often thought of as interpolation. However, more accurately, the process of interpolation is fitting a continuous function to the discrete points in the digital image. This continuous function can then be sampled at whatever points are necessary. In implementing resampling, interpolation and sampling are often combined so that the signal is interpolated at only those points which will be sampled (Fig. 2).

Implementation of resampling from a discrete function to one point in the resampled function is shown in Fig. 3. The interpolating function is shifted so that its peak is aligned with the position of the resampled point. The value of the resampled point is equal to the sum of the values of the original function scaled by the corresponding values of the interpolating function.

Resampling to a larger matrix is often used prior to display in order to make the appearance of the display more pleasing. Increasing the matrix size by resampling *cannot* increase the resolution of the image or the information in the image. (Image processing can only reduce the information in the image.) The purpose of the resampling is to reduce the high-frequency artifacts due to the step changes in intensity that occur at the boundary of each pixel. The resampled image allows the human observer to more accurately appreciate the data since the edge effects at the pixel transitions are removed, or at least moved to a high enough spatial frequency where the eye is less sensitive.

Interpolation With an Ideal Low-Pass Filter

A signal can be exactly reconstructed from samples if the signal is band limited and the sampling is done at a frequency above the Nyquist frequency. Note, however, that unlike the continuous signal, the *sampled* signal is not band limited. Sampling can be viewed as replicating the frequency spectrum at multiples of two pi times the sampling frequency. Interpolation, in contrast, is the opposite of sampling. It produces a continuous signal from a discrete signal. In order to reproduce a band-limited function from a set of samples, the interpolating function should be an ideal low-pass filter. An ideal low-pass filter removes the replicates of the frequency spectrum introduced by the sampling.

A discrete function can be considered to be an exact representation of a band-limited continuous function in the sense that the original function can be reproduced from it. Resampling the discrete function using an ideal low-pass filter for interpolation will produce a new discrete function which is again an exact representation of the original function. Furthermore, if the resampled function is resampled back to the original coordinate points, the original discrete function will be exactly reproduced assuming that the various samplings are all above the Nyquist sampling rate. This argument suggests that the interpolating function which should be used for resampling is an ideal low-pass filter.

There are, however, practical considerations which make

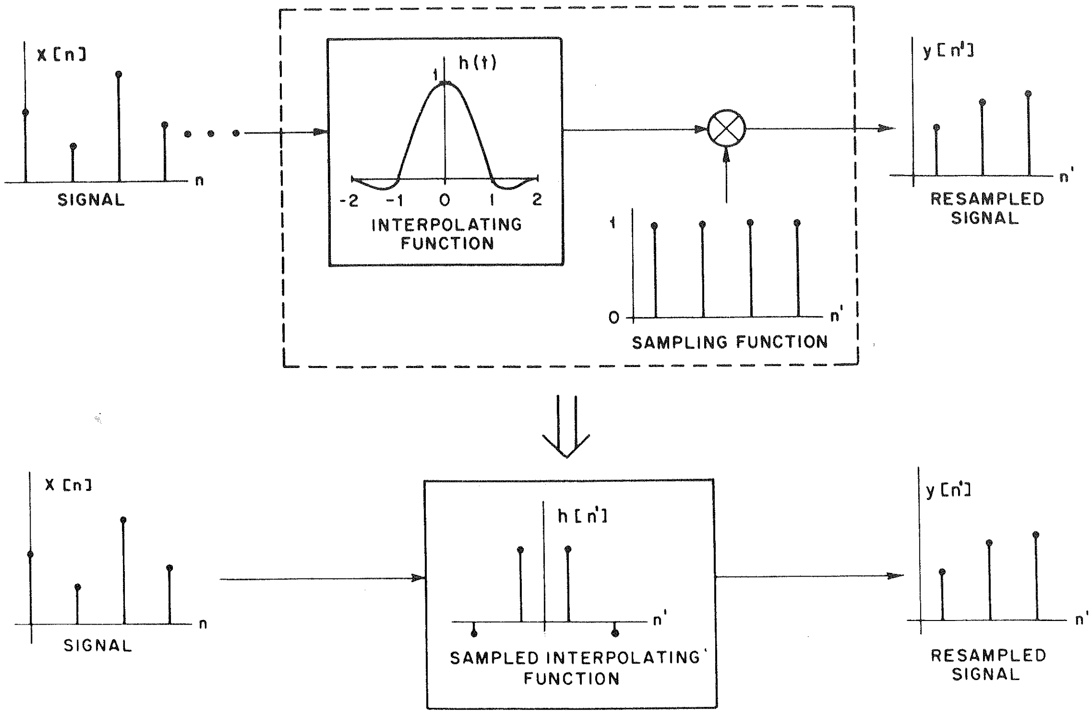


Fig. 2. Implementation of resampling. Resampling can be thought of as interpolation from a discrete to a continuous function followed by sampling (top line). The interpolation is performed by convolving the signal, $x[n]$, with a continuous interpolating function, $h(t)$. The sampling is performed by multiplying the interpolated signal by a comb function. This process is, however, usually implemented by convolving with a sampled interpolating function, $h[n]$ (bottom line). Convolving with a sampled interpolating function is equivalent to interpolating and then sampling.

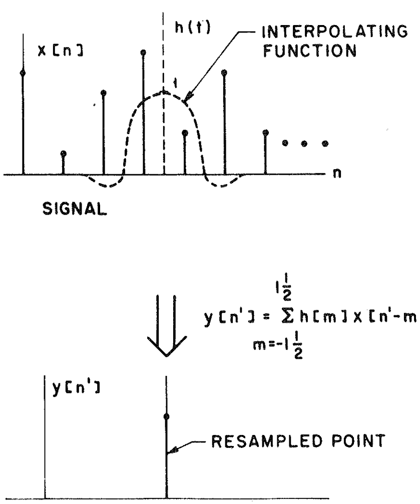


Fig. 3. Resampling of a single point. A single point in the output is the sum of the product of the original signal, $x[n]$, times the value of the interpolating function, $h(t)$, shifted to the location of the output point.

this theoretical resampling technique difficult in the context of picture processing. First, pictures are of finite extent. Therefore, this theoretical description which assumes that the images are of infinite extent is only an approximation; and there will be variations from the theoretical results, especially at the edges of the images. Second, because of the computational burden, the filtering usually takes place in the image domain by convolving with finite impulse response filters of a short duration. For these reasons, an exact interpolation cannot be performed; and consideration must be given to tradeoffs between exact interpolation and computational efficiency.

Interaction Between Interpolation and Sampling

Sampling the interpolated image is equivalent to interpolating the image with a sampled interpolating function (Fig. 2). When resampling from a smaller matrix size to a larger matrix size, there are several points on the sampled interpolating function. Therefore, the resampled image can be thought of as the original image filtered by the unsampled interpolating function. When resampling to a matrix of about the same size, the sampled interpolation function may have a considerably different frequency spectrum than the unsampled interpolation function. The sampling of the interpolating function aliases the higher frequencies of the interpolating function into the lower frequencies. (In the case of an ideal low-pass filter, there are no higher frequencies so the sampled interpolating function has the same spectrum as the unsampled function from $^{-}$pi to $^{+}$pi.) Because of the aliasing caused by sampling the interpolating function, it is necessary to examine not only the unsampled interpolation function, but also typical sampled interpolation functions.

Criteria for an Interpolating Function

The reason for resampling from a smaller to a larger matrix size is often to make an image more pleasing to a human

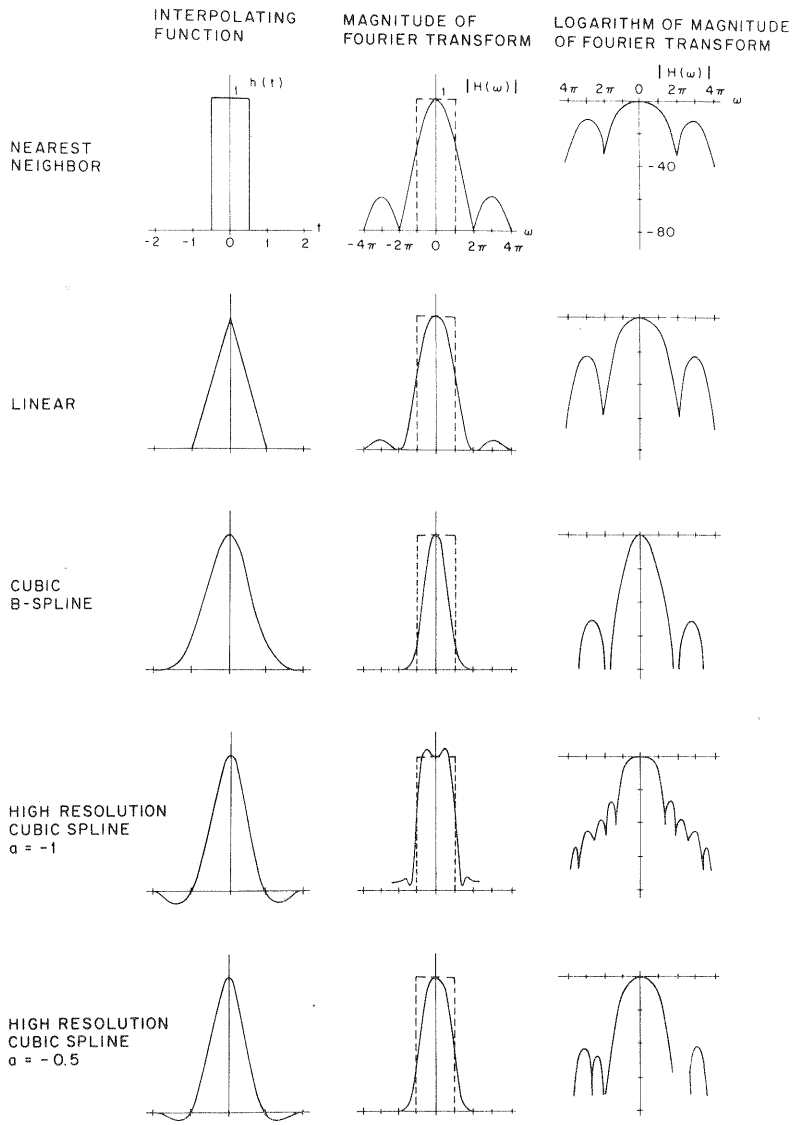


Fig. 4. Interpolating functions. This figure shows several interpolating functions (first coh nn), their Fourier transforms (second column), and the logarithm of their Fourier transforms on an 80 dB scale (third column). The dashed box shows and ideal low-pass filter with cutoff at pi times the sampling frequency. The performance in the pass zone can be best appreciated on the linear plot of the Fourier transform; the stop zone performance can best be appreciated on the logarithm of the Fourier transform plot. The best performance (the best approximation to an ideal low-pass filter) is provided by the high-resolution cubic spline function with $a = -0.5$.

viewer. In this circumstance, the properties of the human visual system must be taken into account. Certain types of distortions will be much better tolerated by the observer than other distortions. For example, noise which is correlated with an image is much more noticeable than noise which is uncorrelated with the image. The property which is sought in the final image is not necessarily its mathematical similarity to the original scene, but rather the appearance of similarity, i.e., verisimilitude. Considerable work has been done on interpolation for a human observer (for example, see Ratzel [10]).

Often, the resampled images are produced for further processing by a computer. In this case, verisimilitude is not necessarily the best property. Rather, mathematical similarity is more desirable. The form of the mathematical similarity will depend on the processing which is to be performed. Keys emphasized similarity of the Taylor series expansion of the two signals [9]. Alternately, it may be desirable for the interpolating function to have a flat frequency response. We have emphasized the examination of the frequency response of various interpolating functions in this paper.

Simple Interpolation Functions

From a computational standpoint, the easiest interpolation algorithm to implement is the so-called nearest neighbor algorithm, where each pixel is given the value of the sample which is closest to it. This method interpolates the sampled image by convolving it with a rectangle function (Fig. 4). Convolution with a rectangle function in the spatial domain is equivalent to multiplying the signal in the frequency domain by a sinc ($\sin(x)/x$) function. The sinc function is a poor low-

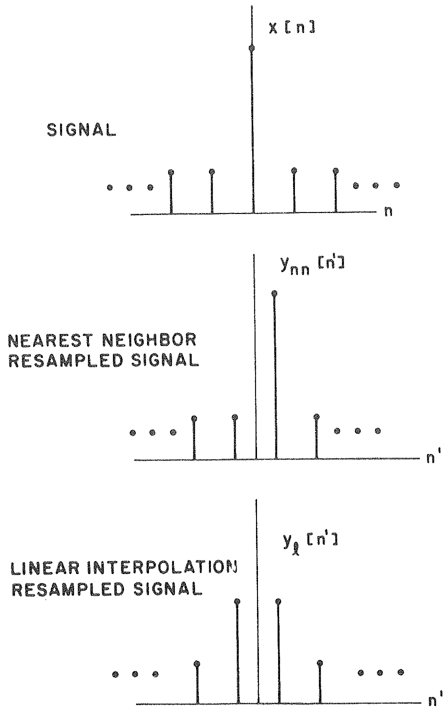


Fig. 5. Effects of resampling. An original signal, $x[n]$, and the same signal resampled with a nearest neighbor, y_{nn}, and linear, y_1, interpolating functions show some of the effects which are produced by resampling. The nearest neighbor algorithm shifts original data to the new coordinate locations. This shift can be as much as $\frac{1}{2}$ pixel. The magnitudes of the pixel values are, however, preserved. The linear interpolation algorithm smooths the data. Notice how the peak in the function has been blunted. The position of the data is, however, not shifted.

pass filter since it has prominent side lobes. Therefore, the nearest neighbor algorithm has a poor frequency domain response.

The nearest neighbor algorithm causes the resampled image to be shifted with regard to the original image by the difference between the positions of the coordinate locations. If, for example, the locations of the resampled points are half way between the original points, the image will be shifted by one-half pixel (Fig. 5). This shift means that the nearest neighbor algorithm cannot be used when it is necessary to preserve sub-pixel image relations. Furthermore, the nearest neighbor algorithm fails completely when resampling to a large matrix size since the pixel values are merely replicated.

There are, however, some quite special properties of the nearest neighbor algorithm. If the resampling is done on a coordinate system with the same spacing as the original coordinate system, then, except for the shift, the resampled data exactly reproduces the original data. That is, the difference between the frequency spectrum of the original and the resampled images is a pure linear phase shift.

A slightly more complicated function to calculate is the linear interpolation algorithm (Fig. 4). Linear interpolation amounts to convolution of the sampled image by a triangle function. This function corresponds in the frequency domain to a modestly good low-pass filter. It does, however, attenuate frequencies near the cut-off frequency resulting in smoothing of the image (Fig. 5), and it does pass a significant amount of energy above the cut-off frequency. Sampling an image which has been bilinearly interpolated will cause the data above the cutoff which has been passed by the interpolating function to be aliased into the low frequencies.

Heuristic Interpolation Function Properties

Since exact interpolation with an ideal low-pass filter can be performed with a sinc function (corresponding to a rectangle function in the frequency domain), this function might seem to be ideal. The problem with the sinc function, however, is that it has considerable energy over an extended distance. Therefore, it cannot be easily implemented as a space domain convolution. It would be natural to try truncating the sinc function over a small distance, but truncation discards a considerable amount of energy. Truncation in one domain leads to ringing in the other domain, so that truncating the sinc function in the space domain will result in ringing in the frequency domain. The ringing will produce undesirable effects. Ratzel tried using a truncated sinc function, but found that it performed poorly compared to other functions such as the cubic spline [10].

The sinc function is, however, useful for developing heuristic properties which are desirable in an interpolating function. A more desirable interpolation function might be a sinc function windowed with a less severe window than the rectangle, e.g., a Hanning window. Such a function should be positive from 0 to 1, negative from 1 to 2, etc. In fact, the functions which have been most successful at preserving high frequencies (see below) tend to have this general shape.

Interpolation Functions Extending Over Four Pixels

The nearest neighbor algorithm interpolates on the basis of a single point. The linear interpolation algorithm interpolates on the basis of the two nearest points. Using three points for interpolation would result in two points on one side of the interpolated point and only one point on the other side; therefore, the next logical interpolation function would use the two nearest points in each direction.

Cubic B-Spline Interpolation Function: Hou and Andrews examined the use of cubic B-splines as interpolation functions. B-splines are several convolutions of the rectangular function. The cubic B-spline is four convolutions of the simple rectangular function (Fig. 4). The B-splines have an extent which is appropriate for interpolation over the two nearest neighbors in each direction [5]. They are reasonably good low-pass filters. However, they are positive in the whole interval from 0 to 2; therefore, they smooth somewhat more than is necessary below the cut-off frequency. They do have good efficiency in the stopband (see results). Since cubic B-splines are symmetric, they only need to be defined in the interval (0, 2). Mathematically, the cubic B-spline can be written

	interval
$f(x) = x^3/2 - x^2 + 4/6$	(0, 1)
$f(x) = -x^3/6 + x^2 - 2x + 8/6$	(1, 2).

When resampling using an interpolating function which extends over four points, if the nearest point is at an offset d, where d is in the interval $(0, 1)$, four samples of the interpolating function will be used: $f(d), f(1-d), f(1+d)$, and $f(2-d)$ (Fig. 1). For the cubic B-spline, the sum of the values of these four points for any value of d is 1. Namely, for any offset the sum of the sampled interpolation function points is equal to 1. Therefore, the dc amplification, the gain on the interpolating function, will be unity. This remarkable property is quite valuable when resampling is done because of geometric distortion or rotation. In these cases, the offset is different at different points in the image. Without this property, the value of a point in the interpolated image would depend upon the alignment of the initial and final coordinates.

General Cubic Spline Interpolation Functions: The cubic B-splines are one type of cubic spline function. More generalized cubic splines have also been considered for interpolation over the two nearest neighbors. By definition a spline is piecewise continuous function; a cubic spline is a piece-wise continuous third-order function. In order to choose from the large number of cubic spline functions, several constraints must be imposed. Like the cubic B-splines (see above), the function should be symmetric about zero; therefore, we need only consider the interval from 0 to 2. The general cubic spline is given by

$$\begin{array}{ll} & \text{interval} \\ f(x) = a_{30}x^3 + a_{20}x^2 + a_{10}x + a_{00} & (0, 1) \\ f(x) = a_{31}x^3 + a_{21}x^2 + a_{11}x + a_{01} & (1, 2). \end{array}$$

There are some natural constraints for a function which is to be used for interpolation. If the resampling is done on the same matrix as the original data, then the original data should be exactly reproduced. This property requires that the value of the function at location 0 is 1, and the values at locations 1 and 2 are 0. Additional logical constraints are that the function should be continuous at locations 0 and 1, that the slope at locations 0 and 2 should be 0, and that the slope of the spline approaching 1 and leaving 1 should be the same. This defines a total of 7 constraints; however, the equations given above have eight unknowns. Thus, these constraints define the cubic spline interpolation function up to a constant [9]:

$$\begin{array}{ll} & \text{interval} \\ f(x) = (a+2)x^3 - (a+3)x^2 + 1 & (0, 1) \\ f(x) = ax^3 - 5ax^2 + 8ax - 4a & (1, 2). \end{array}$$

As with the cubic B-spline, the sum of the four points—$f(d), f(1-d), f(1+d)$, and $f(2-d)$—is equal to 1 [8]. Therefore, for any offset, this interpolating function has the important property that the dc amplification is unity.

With the constant a negative, the function is positive in the interval 0 to 1 and negative in the interval 1 to 2. As the constant a increases, the depth of the side lobe in the interval 1 to 2 increases. Thus, with the free constant negative, the function is of the general form of a windowed sinc function. Because this function has a better high-frequency performance than the cubic B-spline (see below), we shall refer to this function as a high-resolution cubic spline interpolating function.

The choices for the constant a which have been used are -1 [6]–[8], $-\frac{3}{4}$ [8], and $-\frac{1}{2}$ [8], [9] (Fig. 4). The frequency spectrum with $a = -\frac{1}{2}$ is flat in the low frequencies and then falls off toward the cut-off frequency. Keys selected the constant a by making the Taylor series approximation of the interpolated signal agree in as many terms as possible with the original signal. With $a = -\frac{1}{2}$, any second degree polynomial will be exactly reconstructed by interpolation. Keys also considered the rate at which the approximation to a signal converges to the signal as a function of sampling density. With $a = -\frac{1}{2}$, the error of the approximation goes to zero as the third power of the sampling interval. This interpolating function will exactly reproduce a second degree polynominal.

The second derivatives of the two cubic polynominals [one polynomial defined on $(0, 1)$ and the other on $(1, 2)$] can be made equal at 1 by setting $a = -\frac{3}{4}$ [8]. The properties of this interpolating function are intermediate between the choices of $a = -\frac{1}{2}$ and $a = -1$.

Rifman selected $a = -1$ in order to match the slope of the sinc function at 1 [6], [8]. This choice results in some amplification of the frequencies just below the cut-off frequency. The transition between the passband and the stopband is, however, a little bit sharper. Since high-frequency amplification is often appealing to the eye, this choice of the constant a has frequently been used for image processing where verisimilitude is the criterion of interest.

Two-Dimensional Interpolating Functions

For picture processing, the one-dimensional interpolating functions must be transformed into two-dimensional functions. The general approach is to define a separable interpolation function as the product of two one-dimensional functions [10]. Separability is attractive for implementation. Except for the Gaussian function, separability implies that the interpolating function is not isotropic. However, the data on a rectilinear coordinate system are also not sampled isotropically.

METHODS

Interpolating Functions

In order to study the frequency domain characteristics of several candidate interpolating functions, programs were written to allow entry of the interpolating functions and to allow Fourier transformation of these functions. The curves were entered in the Gamma-11 nuclear medicine system curve format (Digital Equipment Corporation) so that they could be displayed with vendor supplied software. Software was written in the RATFOR programming language [11]. The interpolating functions examined were A) nearest neighbor, B) linear interpolation, C) cubic B-spline 5, D) high-resolution cubic spline with $a = -1$ [6], and E) high-resolution cubic spline with $a = -0.5$ [9].

In order to investigate the effect of sampling on the interpolating functions, the number of samples in the function was a variable to the function generating program. To assess the effect of the registration of the original and the resampled coordinate systems, a subsample offset was allowed.

Image Interpolation

The effect of the high-resolution cubic spline interpolating function with $a = -0.5$ was compared to the more common bilinear interpolation algorithm using an image from a coronary arteriogram and an image of an eye. The coronary angiogram was unsharp masked in order to enhance the edges of the vessel. The images were rotated 9 degrees counterclockwise from their original positions. The images were then rotated back to the original position.

RESULTS

Interpolating Functions

The frequency spectra of several of the common interpolation techniques were examined. Fig. 4 shows the space domain representation of several interpolating functions, along with the magnitude of their Fourier transforms both on linear and logarithmic scales. The linear scale most clearly shows the pass zone performance; the logarithmic scale (80 dB) most clearly shows the stop zone performance. These data are similar to the data presented by Ratzel [10] and Keys [9], except that the effect of the constant a in the high-resolution cubic spline has been included.

The space domain representations of the interpolating functions are shown in the first column. From top to bottom they are the nearest neighbor, linear interpolation, cubic B-spline, high-resolution cubic spline with $a = -1$, and high-resolution cubic spline with $a = -0.5$. The second column shows the magnitudes of the Fourier transforms of the interpolating functions. The dotted box shows an ideal low-pass filter with cut-off frequency at pi times the sampling frequency. The third column again shows the magnitudes of the Fourier transforms, but in this case an 80 dB scale has been used.

The high-resolution cubic spline functions have the best response in the pass zone. With the parameter $a = -0.5$, the response is flat at the intermediate frequencies. With $a = -1$, a small amount of amplification of the frequencies just below the cutoff is traded for a more rapid transition between the pass zone and the stop zone. The nearest neighbor function has a reasonable response in the pass zone, although it does have some attenuation even at very low frequencies. Both the linear interpolation and the cubic B-spline have poorer response in the pass zone as would be expected from their smoothing properties.

The high-resolution cubic splines have good response in the stop zone as does the cubic B-spline. The linear interpolating function has poor stop zone performance, and the nearest neighbor has very poor stop zone performance. This poor stop zone performance means that resampling after interpolation with either of these latter two functions will result in a large amount of aliasing.

In summary, the nearest neighbor function does moderately well in the pass zone, but very poorly in the stop zone. The linear interpolation function performs better in the stop zone, but at the expense of a considerable amount of smoothing in the pass zone. The cubic B-spline has quite good stop zone performance, but does the most smoothing in the pass zone. The high-resolution cubic splines have the best combination of pass zone and stop zone performance. A flat pass zone response and good stop zone performance are obtained with a value of $a = -0.5$, while $a = -1$ trades some high-pass filtering and a somewhat poor stop zone response for better preservation of the frequencies near the cutoff.

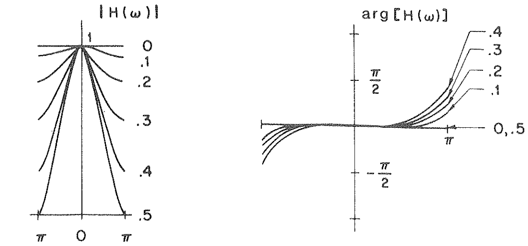


Fig. 6. Sampled high-resolution cubic spline interpolation function. The sampled interpolating function will alias the frequencies in the stop zone into the pass zone resulting in considerably different performance than the unsampled interpolating function. The magnitudes, $H(\omega)$, and phases, $\arg(H(\omega))$, of the high resolution cubic spline function ($a = -0.5$) are shown for offsets 0, 0.1, 0.2, 0.3, 0.4, and 0.5. At 0 offset, the sampled interpolating function is an ideal low-pass filter. As the offset increases, the interpolating function deviates from an ideal low-pass filter.

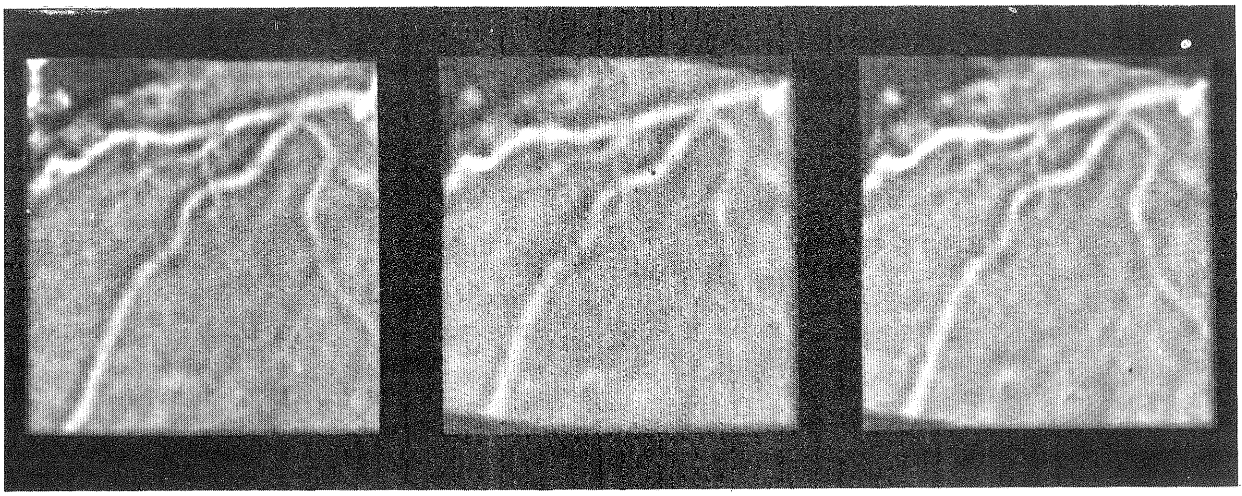
Sampled Interpolating Functions

In order to show the effects of the registration of the original and the resampled coordinate systems, sampled interpolating functions were examined. The sampling corresponds to resampling to a matrix with the same size as the original matrix, but with a subpixel translation. Translations of 0, 0.1, 0.2, 0.3, 0.4, and 0.5 pixels were used. The resampling will alias any data which are passed in the stop zone into the pass zone. The translation of the resampled matrix points with respect to the original matrix points will determine the phase with which these aliased data are added to the pass zone.

Fig. 6 shows the results using the high-resolution cubic spline function ($a = -0.5$). At zero translation, the magnitude of the Fourier transform is everywhere equal to one due to the definition of the high-resolution cubic spline (the value at 0 is 1, and the value at 1 and 2 is 0). The response falls off going from 0 translation to 0.5 translation. At the 0.5 translation the magnitude is zero at pi. The magnitude of the dc response is one for all translations.

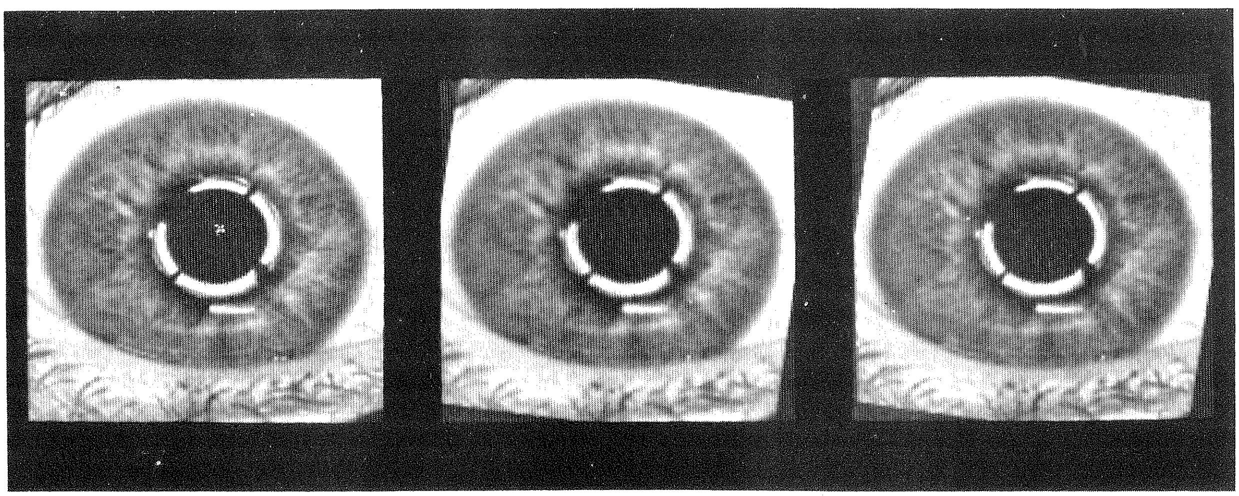
Thus, the resampling has a dramatic effect on the response of the interpolating function. At 0 translation, the resampled response is improved with respect to the interpolated response. Initially it might seem surprising that the aliased data improves the response with respect to the interpolated data. However, remember that one criterion for the interpolating function was that for resampling at the same points, the initial function would be exactly reproduced. At 0.5 translation, the sampled response is considerably worse than the interpolated response.

Fig. 6 also shows the phases of the high-resolution cubic



(a) (b) (c)

Fig. 7. Image resampling. (a) Initial image of a coronary angiogram. The primary data is 64 × 64 with a display dimension of 128 × 128. (b) Resampling using the bilinear interpolating algorithm. Notice the loss of sharpness at the edges of the vessels. (c) Resampling using the high-resolution cubic spline.



(a) (b) (c)

Fig. 8. Image resampling. (a) Initial image of an eye of a blue-eyed subject. The data and display dimensions are 128 × 128. (b) Resampling using the bilinear interpolating algorithm. Notice the loss of detail in the iris. (c) Resampling using the high-resolution cubic spline.

spline functions ($a = -0.5$) for translations of 0, 0.1, 0.2, 0.3, 0.4, and 0.5. Translations 0 and 0.5 have zero phase shifts; the other translations have some phase shift at the higher frequencies. The space domain representation of the interpolation functions suggests why there may be some phase shift. The point nearest to the location of the interpolated point is weighed most heavily. Although the effect is small, for translations less than 0.5 pixels, the highest frequencies do not adequately reflect the translation.

Image Interpolation

Figs. 7 and 8 show the results of resampling an image with linear interpolation and with the high-resolution cubic spline. There was only a small loss of detail in the image which is rotated using the high-resolution cubic spline algorithm, but there is considerable smoothing using the bilinear interpolation. In the angiogram (Fig. 7) the edge of the vessels is less distinct using bilinear interpolation. In the image of the eye (Fig. 8), there is loss of the iral detail with the bilinear interpolation.

Discussion

The nearest neighbor algorithm has the shortest extent in the image domain, one interpixel distance. It does well in the pass zone, but it has very poor stop zone response. The linear interpolating algorithm, which extends over two interpixel distances, has a somewhat better overall response. The high-resolution cubic splines, which extend over four interpixel distances, have better response in both the pass zone and the stop zone. This improvement in response with filter length is just what would be expected from design of a limited extent filter.

The choice between the different length four interpolating functions depends upon the task. The cubic B-spline provides the most smoothing. The $a = -1$ high-resolution cubic spline provides the best high-frequency response along with some high-frequency enhancement. These characteristics may be most appropriate for producing verisimilar imagery. The $a = -0.5$ high-resolution cubic spline has both a flat low-frequency response and good stopband performance. These characteristics may be most appropriate when further mathematical processing of the images is to be performed.

Sampling the interpolated image results in aliasing of the frequencies above pi times the sampling frequency. For the high-resolution cubic spline functions, there is a special case when there is an exact match between the initial sampling coordinates and the resampling coordinates. In this case the aliasing results in an exact reconstruction of the image. That the aliased frequencies from the stop zone exactly replace the attenuation in the pass zone may at first seem surprising. However, this unusual property comes from one of the image domain criteria used to define these interpolation functions, namely, that when the resampled point exactly corresponds to a point in the original image, then it should exactly reproduce that point.

Resampling midway between the initial images results in the frequencies at pi being added in with a phase shift of 180° Thus, the worst problems with aliasing occur when resampling midway between the initial sample points. Fig. 6 shows the response of the sampled interpolating function for different offsets between the initial and the resampled coordinate locations. The relationship between the location of the initial samples and the location of the resampled points often has a greater effect on the system response than the choice between the interpolation function used.

The visual results of interpolation with the $a = -0.5$ high-resolution cubic spline are quite good as compared to the more common bilinear interpolation algorithm. There is less smoothing of the high-resolution detail in the image. It does require additional computation; however, the improvement in the interpolation may be worth the computational burden for several tasks. Although the length for interpolating functions is adequate for many tasks, additional improvement in performance could be made by including more points in the calculation. The same criteria for selecting interpolating functions used in this paper—frequency response, phase shift, and gain—would also be useful in selecting between functions with a longer extent.

Acknowledgment

The authors wish to thank Dr. A. Arott and Dr. R. Spears, for providing the images which were used for analysis.

References

[1] P. E. Anuta, "Spatial registration of multispectral and multitemporal digital imagery using fast Fourier transform techniques," *IEEE Trans. Geosci. Electron.*, vol. GE-8, pp. 353–368, 1970.

[2] P. van Wie and M. Stein, "A Landsat digital image rectification system," *IEEE Trans. Geosci. Electron.*, vol. GE-15, pp. 130–137, 1977.

[3] S. J. Riederer and R. A. Kruger, "Intravenous digital subtraction: A summary of recent developments," *Radiology*, vol. 147, pp. 633–638, 1983.

[4] R. A. Kruger, F. J. Miller, J. A. Nelson, P. Y. Liu, and W. Bateman, "Digital subtranction angiography using a temporal bandpass filter: Associated patient motion properties," *Radiology*, vol. 145, pp. 315–320, 1982.

[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] S. S. Rifman, "Digital rectification of ERTS multispectral imagery," in *Proc. Symp. Significant Results Obtained from the Earth Resources Technology Satellite-1*, 1973, vol. I, sec. B NASA SP-327, pp. 1131–1142.

[7] J. E. Taber, "Evaluation of digitally corrected ERTS images," in *Proc. Symp. Third Earth Resources Technology Satellite-1*, 1973, vol. I. sec. B NASA SP-351, pp. 1837–1843.

[8] K. W. Simon, "Digital image reconstruction and resampling for geometric manipulation," in *Proc. IEEE Symp. Machine Processing of Remotely Sensed Data*, 1975, pp. 3A-1–3A-11.

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

[10] J. N. Ratzel, "The discrete representation of spatialy continuous images," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, 1980.

[11] R. G. Beckmann, M. S. Sasaki, H. D. Royal, J. A. Parker, and G. M. Kolodny, "Incorporation of a RATFOR preprocessor into a nuclear medicine computer system," in *Digital Imaging: Clinical Advances in Nuclear Medicine.* New York: Society of Nuclear Medicine, 1982, pp. 165–172.