The Constant MTF Interpolator

A Resampling Technique with Minimal MTF Losses

Ellis Freedman
Serious Science, LLC
Meadowbrook, PA USA
ellis.freedman@serioussciencellc.com

Dr. Robert Fleming
North Wales, PA USA

***Abstract*— The geometric remapping of pixel values during the processing of digital imagery, such as magnification, warping and registration, can significantly affect the final image quality. Many medical imaging systems include a resampler/interpolator, such as bicubic, as part of their processing, that acts as a variable low pass filter. This not only degrades the spatial frequency response of the image and increases blurring, but varies the degradation with the fractional pixel interpolation distance. Thus the final modulation transfer function (MTF) of the image is often degraded in an almost random pattern for which accurate compensation cannot be applied. This reduces image interpretability and degrades pixel intensity accuracy. To counter this, a replacement table of interpolation kernels has been developed that imposes virtually the same MTF degradation for any pixel interpolation distance. The inverse MTF can then be applied such that there is minimal MTF degradation to the image after interpolation. This paper provides the latest results of analyses and simulations comparing the performance of the improved Constant MTF Interpolator (CMTF) to the cubic interpolator. These results indicate that the CMTF provides dramatically better image quality, preserves pixel intensity fidelity, and enhances the placement accuracy of pixels in the resampled imagery more than other tested interpolators.**

Keywords—interpolation, resampling, modulation transfer function, MTF

I. Introduction

Many imaging systems geometrically remap pixels for warping, registration, magnification, or other purposes. Each of these remappings require some form of resampling and interpolation to estimate values between the original pixel locations. All of the techniques that are used to perform resampling act as low pass spatial filters with bandpasses that vary with interpolation distance [1][2]. This results in a reduction of the modulation transfer function (MTF) and blurring of the image that often varies in intensity from pixel to pixel.

To counter this effect, a new interpolation technique, the Constant Modulation Transfer Function (CMTF) interpolator, has been developed with the goal of minimizing the variations in the low pass filter bandpasses between interpolations distances. Since the filtering impact is always the same, a single, fixed, inverse filter can be applied to the image so that there will be no net blurring effect on the final, resampled image.

This paper will review one of the most commonly used interpolation techniques, the cubic convolution, explain the CMTF alternative approach, and describe the differences in their impacts to the image MTF and their respective geometric accuracies. Finally, visual comparisons are provided between images resampled with the two techniques.

II. Cubic Convolution

The cubic convolution is one of the most common interpolators in use. The kernel is created from piecewise cubic polynomials defined over the subintervals (-2,-1), (-1,0), (0,1) and (1,2) with constraints on the values and derivatives at the knots to ensure continuity. This forms eight degrees of freedom and seven equations that result in equation (1) [3]. The parameter, α, is the slope at x=1, and is typically assigned the value -0.5. For each fractional interpolation distance the equation provides a four element convolution kernel. When convolved with four consecutive data points/pixels, a new, interpolated value is provided between the second and third data point/pixel.

$$h(x) = \begin{cases} (\alpha+2)|x|^3 - (\alpha+3)|x|^2 + 1 & 0 \le |x| < 1 \\ \alpha|x|^3 - 5\alpha|x|^2 + 8\alpha|x| - 4\alpha & 1 \le |x| < 2 \\ 0 & 2 \le |x| \end{cases} \tag{1}$$

The convolution kernels can be calculated and applied from equation 1 for each pixel's interpolation distance on a pixel by pixel basis during the resampling process, or a table of interpolation kernels can be created once, prior to processing, with quantized interpolation distance increments, and the convolution kernel determined by a simple table lookup to the nearest interpolation distance.

While the cubic convolution is regarded as more accurate than less robust interpolators such as nearest neighbor and linear, and it has been described as a compromise between accuracy and computational intensity, it exhibits the same low pass filtering issues as all other interpolators. Its spatial frequency response varies significantly with interpolation distance. The effective MTF for the cubic convolution as a function of interpolation distance is provided in Figure 1. For

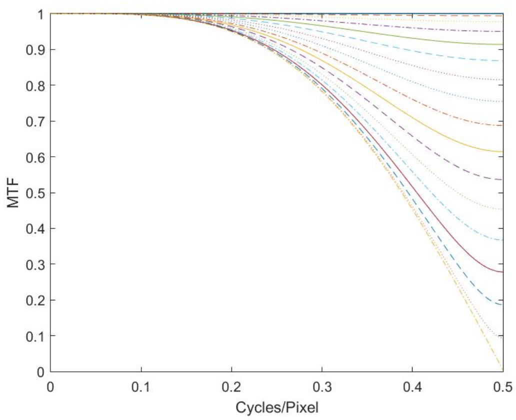


Figure 1. MTF of cubic convolution for interpolation distances 0 - 1 pixels in increments of 1/32 of a pixel.

most resampling processes, this results in a blurring in a pseudorandom manner across the image. And because the blurring varies from pixel to pixel, there is no way to compensate for this effect without either oversharpening or undersharpening the image. Therefore, the viewer will experience unknown losses in image detail.

III. The Constant MTF Interpolator

The Constant MTF interpolator is similar to the cubic convolution in many ways. It is derived from a series of constraints, just as with the cubic convolution, and kernels can be derived for any interpolation distance. It is constrained to provide an exact solution for constant, linear and quadratic fits. However, unlike the cubic convolution, the kernels consist of six elements rather than four, the differences between the MTF at all frequencies for every interpolation distance have been minimized, and the MTF value for all distances at Nyquist is 0.3. This provides two major advantages over other interpolators:

- Because the MTF is virtually the same for all interpolation distances, imagery does not experience pseudorandom blurring from pixel to pixel.

- Because the MTF never approaches zero and is nearly identical for all interpolation distances, a single, inverse filter can be applied to create an overall MTF that is close to one at all frequencies and effectively eliminates blurring caused by the interpolation process.

IV. Derivation of the CMTF

The goal for the derivation of the CMTF is to create an interpolator in the form of a set of convolution kernels such that the resulting MTF for any interpolation distance is virtually the same, and that the error in placing a perfect edge at the new interpolation distance is minimized.

Begin by choosing the size of the kernel to be six elements, and then impose constraints on the form of the kernels to reduce the degrees of freedom and aid in finding the solution.

The form of the interpolator as a convolution is,

$$\hat{f}(p) = \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} c(n,p) f(x_n) \tag{2}$$

where,

$\hat{f}(p) = interpolated\ value\ of\ f(x)\ at\ p\ above\ n = 0$
$N = size\ of\ kernel = 6$
$n = samples\ points$
$c(n,p)$
$= coefficients\ designed\ to\ interpolate\ distance\ p$
$f(x) = function\ to\ be\ interpolated$
$x_n = value\ of\ x\ sampled\ at\ point\ n$

Assume that the Discrete Fourier Transform (DFT) is defined by,

$$\hat{c}(u,p) = \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} c(n,p) e^{-i2\pi un} \tag{3}$$

where,

$\hat{c}(u,p) = coefficients\ of\ DFT\ of\ c(n,p)$
$u = spatial\ frequency$

And that the MTF is the magnitude of the DFT.

$$MTF(u,p) = \sqrt{\hat{c}(u,p) \cdot \hat{c}^*(u,p)} \tag{4}$$

The interpolator, which acts as a filter, shall have no gain, and provide a perfect interpolation value for a constant function. A constant function implies no other frequency content except at a frequency of zero, the DC value. Therefore, define the DFT to have a value of one at a frequency of zero.

$$F(0) = 1 = \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} c(n) e^{-i2\pi n \times 0} = \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} c(n) \tag{5}$$

The sum of the coefficients of the kernel is, therefore, one.

The interpolator should also ensure an exact interpolation solution, f(p)=mp+b, for the linear function, f(x)= mx+b.

$$\hat{f}(p) = mp + b = \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} c(n,p)(mn + b) \tag{6}$$

which becomes,

$$p = \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} c(n,p)(n) \tag{7}$$

Just as with the linear solution, the interpolator should provide an exact solution for the quadratic function, $f(p)=Ap^2+Bp+C$, for the function, $f(x)=Ax^2+Bx+C$.

$$\hat{f}(p) = Ap^2 + Bp + C \tag{8}$$

$$Ap^2 + Bp + C = \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} c(n,p)(An^2 + Bn + C) \tag{9}$$

By simply multiplying through by all values of n (in this case, -2, -1, 0, 1, 2, and 3) this simplifies to,

$$\hat{f}(p) = p^2 = \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} c(n,p)n^2 \tag{10}$$

Finally, ensure that every kernel of the interpolator provides the same MTF value at Nyquist (u=0.5).

$$\hat{c}(0.5,p) = \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} c(n,p)e^{-i2\pi n\times 0.5} \tag{11}$$

But since this has no imaginary components,

$$MTF_N(p) = \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} c(n,p)(-1)^n \tag{12}$$

At this point there are four linear constraints that can be written in the matrix form, $A * C = B$

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 & 3 \\ 4 & 1 & 0 & 1 & 4 & 9 \\ 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \tag{13}$$

$$C = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} \tag{14}$$

$$B = \begin{bmatrix} 1 \\ p \\ p^2 \\ MTF_N \end{bmatrix} \tag{15}$$

which provides four equations and six unknowns. In theory, a search could be performed in the space of the two unknown coefficients for a solution that minimizes differences between the MTFs of the kernels and placement error. In practice, however, there are many local minima that will provide undesirable solutions.

However, a non-linear constraint can further reduce the solution space. Recall that the purpose of the CMTF is to ensure that the resulting MTF is the same for all interpolation distances. From equation (3), the DFT of the kernels can be rewritten in matrix form as,

$$[\hat{c}(u,p)] = [c(n,p)] \cdot T \tag{16}$$

where,

$$T = matrix\ with\ column\ elements\ e^{-i2\pi u\left(-\frac{N}{2}+1\right)}$$
$$to\ \ e^{-i\pi uN}\ and\ u\ varying\ by\ column$$

In order to force the MTFs, which are the magnitudes of the DFT of the kernels, to be the same, impose a minimization constraint on the sum of the squares of the differences of the MTFs at all frequencies.

$$min \sum_{j\neq k} \sum_{i=1}^{N} \left[MTF(u_i,p_j) - MTF(u_i,p_k)\right]^2 \tag{17}$$

$$j,k = Set\ of\ all\ desired\ interpolation\ distances$$
$$N = Number\ of\ frequencies\ in\ DFT\ of\ kernels$$

There are now five equations and six unknowns. The final constraint is the minimization of the placement error. Placement error is defined as the absolute difference between the intended interpolation distance and the interpolation distance achieved when applying the interpolation kernel to an ideal edge (i.e. a step function such the f(x) =0 for x<0, f(x)=1 for x≥0, sometimes called the Heaviside Function.)

The Fourier Transform of the ideal edge described above is given by,

$$\hat{E}(u) = \int_{-\infty}^{\infty} E(x)e^{-i2\pi ux}dx \tag{18}$$

$$\hat{E}(u) = \frac{\delta(u)}{2} + \frac{i}{2\pi u} \tag{19}$$

The system response to applying the Fourier Transform of an interpolation kernel is therefore,

$$S(u) = \hat{E}(u)MTF(u)e^{i\,\varphi(u)} \tag{20}$$

where,

$$MTF(u) = magnitude\ of\ the\ transform\ of\ the\ kernel$$
$$\varphi(u) = phase\ of\ the\ transform\ of\ the\ kernel$$

Therefore, the position of the edge after application of the kernel can be written with the inverse Fourier transform as,

$$E(x) = \int_{-0.5}^{0.5} \hat{E}(u)MTF(u)e^{i\,\varphi(u)}e^{i2\pi ux}du \tag{21}$$

The limits are set to the width of the periodic function.

After some manipulation, this becomes,

$$E(x) = \frac{1}{2} + \frac{1}{2\pi}\int_{-0.5}^{0.5} \frac{MTF(u)\sin(2\pi ux - \varphi(u))}{u}du \tag{22}$$

The value of x in this equation that forces the integral to zero is the position of the interpolated edge. Although unwieldy, a simple search can determine this value to several decimal places. Thus the placement error is the absolute value of the difference of the desired placement for this kernel and the position determined from minimizing value of x.

In order to begin a search for the CMTF kernels that meet these requirements, we need a starting point so that an exhaustive search need not be performed. A relatively simple starting point is a two point filter. The filter will have an MTF value of one at zero frequency, and MTF_N at Nyquist. To determine the MTF for this filter, we first determine the DFT of the filter.

$$F(c(n)) = \sum_{n=0}^{1} c(n) e^{-i2\pi un} \quad (23)$$

$$F(c(n)) = c(0) + c(1)e^{-i2\pi u} \quad (24)$$

$$F(c(n)) = c(0) + c(1)\{\cos(2\pi u) - isin(2\pi u)\} \quad (25)$$

$$MTF_{2\,pt}^2(u) = F(c(n))F^*(c(n)) = [c(0) + c(1)\{\cos(2\pi u) - isin(2\pi u)\}][c(0) + c(1)\{\cos(2\pi u) + isin(2\pi u)\}] \quad (26)$$

$$MTF_{2\,pt}^2(u) = c^2(0) + c^2(1) + c(0)c(1)\{\cos(2\pi u) - isin(2\pi u)\} + c(0)c(1)\{\cos(2\pi u) + isin(2\pi u)\} \quad (27)$$

$$MTF_{2\,pt}^2(u) = c^2(0) + c^2(1) + 2c(0)c(1)\cos(2\pi u) \quad (28)$$

If we set F(c(n))=1.0 for u=0, and F(c(n))=MTF_N at u=0.5 (Nyquist), then,

$$c(0) = \frac{(1+MTF_N)}{2} \quad (29)$$

$$c(1) = \frac{(1-MTF_N)}{2} \quad (30)$$

and the initial shape of the MTF curve becomes,

$$MTF_{2\,pt}^2(u) = \left\{\frac{(1+MTF_N)}{2}\right\}^2 + \left\{\frac{(1-MTF_N)}{2}\right\}^2 + \frac{(1-MTF_N{}^2)}{2}\cos(2\pi u) \quad (31)$$

We can now create a six element kernel as a starting point for the search with N=6 and,

$$\hat{c}_{2\,pt}(n) = \sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}} MTF_{2\,pt}(u) e^{i2\pi un} \quad (32)$$

Providing this starting point and the constraints described above to a routine such as Matlab's fmincon results in a set of kernel coefficients that all provide similar MTF impacts and minimize the final placement errors. Depending upon the initial MTF curve, the form of the non-linear constraints and the chosen value at Nyquist, results may be improved by recursively running the routine using the MTF of the output curves as initial MTF inputs.

Figure 2 shows plots of the MTFs of the CMTF interpolator for selected interpolation distances both before and after inverse filter compensation. Note that after compensation, the lowest value for any MTF and any interpolation distance is about 0.95, while with the cubic convolution, the MTF can drop to as low as zero.

Figure 3 shows a comparison of the placement error for both the cubic convolution and the CMTF for interpolation distances from zero to one pixel in increments of 1/32 of a pixel. Note that one of the constraints of the cubic convolution forces the placement error to zero at integer and half integer distances. The CMTF does not use these constraints. While the CMTF has a lower average placement error across all distances, neither method displays significant placement error.

V. EFFECTS ON IMAGERY

In order to provide examples of the effects of both convolution techniques, single slice images from cranial MRIs were chosen as test images. Figure 4 shows three inset images. The top image is the original image. The middle image is the difference between the original image and an image that was resampled with the cubic convolution 15/32 pixels to the right and down and then resampled again 15/32 to the left and up. The bottom image is the difference between the original image and an image that was resampled with the CMTF 15/32 pixels to the right and down and then resampled again 15/32 to the left and up. Figure 6 shows similar images for a sagittal view with a 13/32 pixel shift in all directions. The double resampling was performed to ensure that the test images were registered to the original image.

The middle image in each example displays the significant losses in detail and edges that can be produced when resampling with the cubic convolution. The lower images display almost no loss in detail, although some very faint edges may be seen. This is because the overall MTF of CMTF is not exactly one at all frequencies, and a small amount of detail can still be lost.

It should also be noted that since the low pass filtering effects of the cubic convolution are much stronger than those of the CMTF, the signal to noise ratio of the cubic resampled image will be larger, because it filters out noise with the edges. The image resampled with CMTF will have approximately the same signal to noise ratio as the original image.

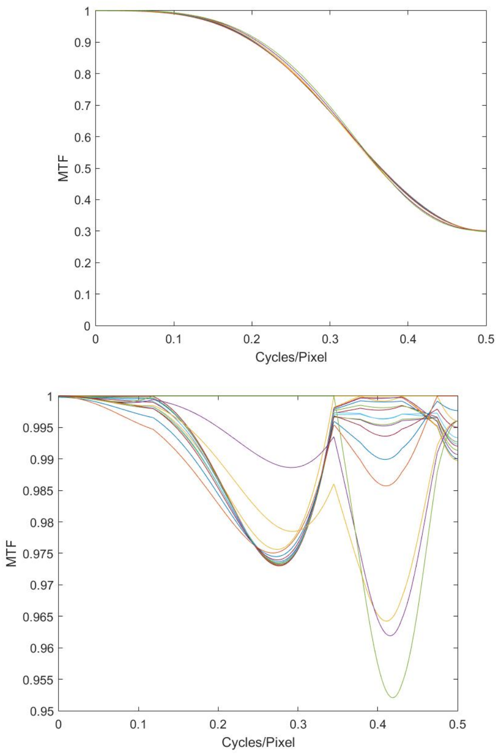


Figure 2. MTF of CMTF for interpolation distances 0 - 1 pixels in increments of 1/32 of a pixel. Top inset shows MTFs prior to compensation. Bottom inset shows MTFs after compensation.

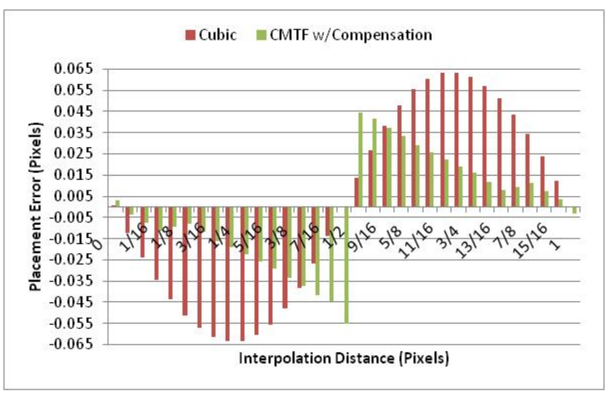


Figure 3. Placement error for cubic convolution and CMTF

A different example of the losses exhibited by the cubic convolution vs. the CMTF can be provided by magnification. The original Figure 4 sample image was magnified by a factor of 2.2 times using the cubic convolution and the CMTF. Magnification by a non-integer amount provides five different interpolation distances distributed throughout the image. Figure 5 shows the difference between the two magnified images. Similarly, the difference between the Figure 6 image magnified 4.12 times with the cubic and the CMTF are shown in Figure 7. Here, too, it is apparent that CMTF preserves the high frequency details lost by using the cubic convolution.

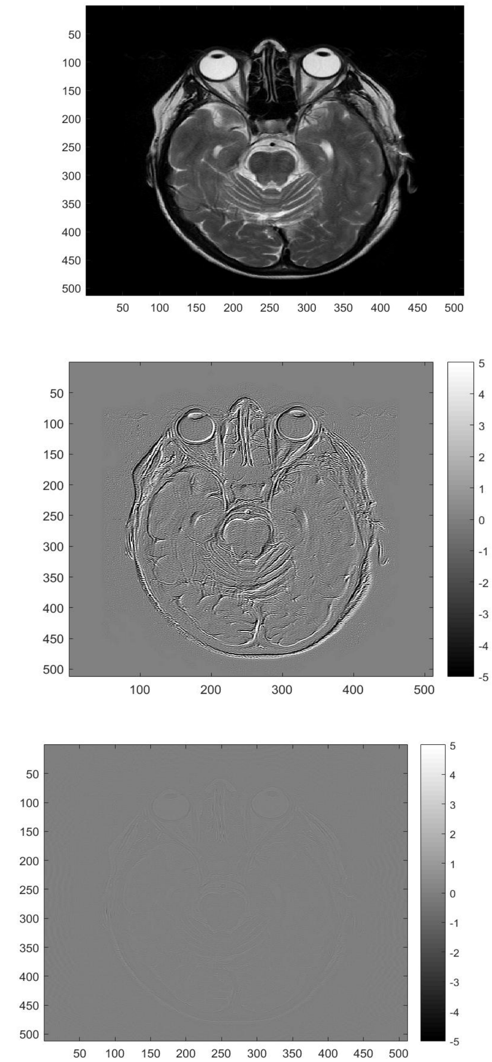


Figure 4: Original image (top), difference between original image and image interpolated with cubic convolution forward and backward 15/32 of a pixel in x and y directions (middle), and difference between original image and image interpolated with CMTF forward and backward 15/32 of a pixel in x and y directions (bottom).

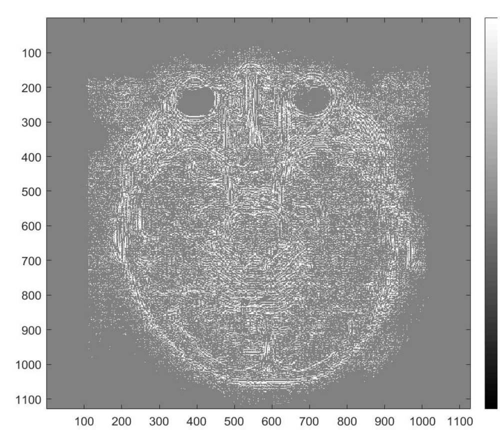


Figure 5. Difference between images magnified 2.2X with cubic and CMTF

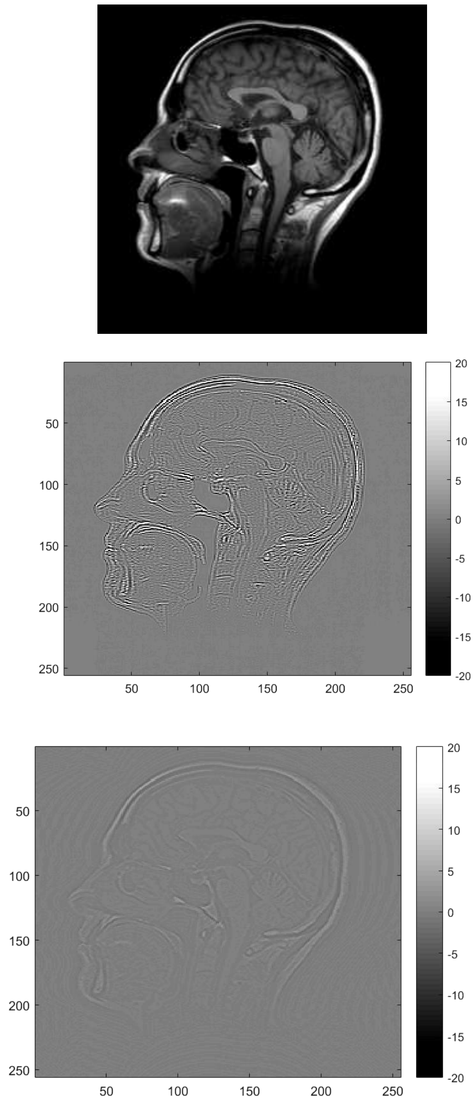


Figure 6: Original image (top), difference between original image and image interpolated with cubic convolution forward and backward 13/32 of a pixel in x and y directions (middle), and difference between original image and image interpolated with CMTF forward and backward 13/32 of a pixel in x and y directions (bottom).

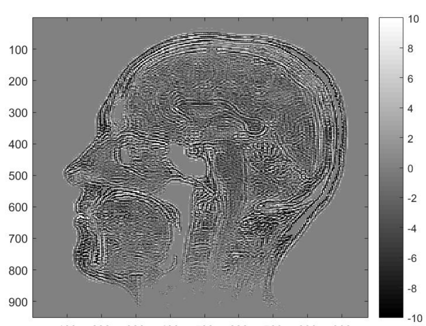


Figure 7. Difference between images magnified 4.12X with cubic and CMTF

VI. Summary

Current image interpolators act as low pass filters that can blur images in pseudorandom and uncompensatable ways. The Constant MTF (CMTF) interpolator was designed to minimize or eliminate the low pass filtering effects and maintain geometric accuracy. It was shown that the CMTF has an effective MTF greater than 0.95 at all frequencies, while the cubic convolution MTF can fall to as little as zero, and that the CMTF has geometric placement accuracy comparable to, and often better than that of the cubic convolution. Images resampled with fixed interpolation distances or through magnification with the cubic convolution show significant loss of detail and edges compared to those resampled with the CMTF.

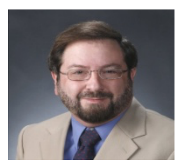
Because resampling is used for so many image processing functions, including warping, registration, magnification and super resolution, the replacement of current interpolators with CMTF should provide significant improvements to the quality of images that undergo these processes.

References

[1] Thomas M. Lehmann, Claudia Gönner, Klaus Spitzer, "Survey: Interpolation Methods in Medical Image Processing," IEEE Transactions on Medical Imaging, vol 18, No. 11, November 1999.

[2] Philippe Thévanaz, "Interpolation Revisited," IEEE Transactions on Medical Imaging, vol 19, No. 7, July 2000.

[3] Robert G. Keys, "Cubic Convolution Interpolation for Digital Image Processing," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-29, No. 6, December 1981



Ellis Freedman received a B.S. degree in Physics from Temple University in 1976, and an M.S. degree in Physics from Northeastern University in 1977. In 1978 he joined Veda, Inc. as a systems engineer. In 1983, he joined GE Aerospace (later Lockheed Martin Co.) as a systems engineer specializing in the calibration and processing of imagery, and the development of mathematical algorithms, modeling tools, and processing techniques for the correction, enhancement, and exploitation of remote sensing imagery. He retired in 2010 after achieving the title of Lockheed Martin Fellow to form his own company, Serious Science LLC.

Robert Fleming received a BA degree in Math-Physics from LaSalle College in 1959, an MS in Mathematics from the University of Notre Dame in 1961, an MS degree in 1968 and a Ph.D. in 1973 in System Science from the Polytechnic Institute of Brooklyn. In 1961 he joined General Electric Aerospace (later Lockheed Martin Co.) as a mathematician specializing in computer applications in fluid mechanics, thermodynamics and orbital application. Later he concentrated on processing and interpretation of satellite data. He retired in 1999 and later became a consultant to Lockheed Martin.