

# Model-based least squares optimal interpolation

A. Gilman, D.G. Bailey, S. Marsland  
 School of Engineering and Advanced Technology  
 Massey University  
 Palmerston North, New Zealand  
 a.gilman@massey.ac.nz

**Abstract**—The traditional approach to image interpolation is by synthesis using basis functions because of its computational simplicity and experience-proven quality of the result. We offer an alternative approach to designing the basis (interpolation kernels), using least-squares optimisation and image models that encompass the prior knowledge. In this paper we consider and derive a finite-support interpolation kernel based on a step-edge model and show that this results in a piece-wise cubic polynomial similar to Keys' cubic convolution. We offer an experimental comparison of the proposed kernel to a number of common methods and show that it performs similar to, or better than, the existing methods with similar extent of spatial support.

*Keywords*—interpolation; resampling; least-squares optimisation

## I. INTRODUCTION

Image interpolation is a crucial procedure in the processing of digital images, due to their discrete (sampled in space) nature. Every time values that fall between the sampling points are required, the continuous function must be reconstructed and then resampled at these new locations. However, in general only the values of the continuous function at the new locations are computed for computational reasons.

For a band-limited function, sampled higher than the Nyquist rate, the continuous function can be reconstructed using sinc-interpolation [1]. Unfortunately, the sinc kernel has infinite extent with a relatively slow decay. This led to a multitude of kernels with finite support that approximate the sinc, see [2] for example. Other traditional approaches include synthesis using basis functions, such as various piece-wise polynomials [3]. These methods are generally one-dimensional and have been applied to the problem of image interpolation in a separable fashion (i.e., treating each spatial dimension separately), e.g. [3, 4]. Recently, attempts have also been made to produce non-separable kernels for image interpolation [5-7]; but, these come at an obvious increase in computational cost.

Images often have components above the Nyquist rate, leading to aliasing of the higher frequencies. Reconstruction of the continuous function becomes an ill-posed problem with infinite number of solutions. Some prior information is required to pick the most plausible one. In traditional methods, this prior information is “encoded” in the basis functions. Other methods, such as the Bayesian framework [8], for example, can make more explicit use of these image priors.

In previous work [9], we explored the idea of a least-squares optimal interpolation kernel. We used simulated images, such that the output is freely available, and optimised a

finite-support kernel by minimising the mean square error with respect to the kernel weights. The resulting optimal kernel would resample another image with near-optimal results and in later work we demonstrated that one-dimensional optimal kernels optimised on different images are quite similar [10]. It was also shown that a similar kernel can be obtained by using a model image. These results were used in [11], where we optimised a non-separable 2D kernel using an image model. However, this work was completed in light of image super-resolution [12] and the methods were specifically designed to resample non-uniformly sampled data on a uniform grid.

In this paper, we adapt this methodology to derivation of 1D kernels for interpolation of uniform images in a separable fashion, much like a piece-wise polynomial kernel would. The novelty of this method lays in the way the kernel is produced from an image model that can encompass the prior knowledge, making it specific to interpolating image data. We use a step-edge model as an example here, but other image models can be employed in a similar way.

## II. LEAST-SQUARES OPTIMAL INTERPOLATION

### A. Optimal Interpolation

One of the main uses of image interpolation is resampling. If values of the image function are required at locations other than the ones where the image is defined, it is necessary to interpolate the image and resample it at the new locations. A simple case of this procedure is considered here: resampling an image  $f[k,l]$  on an offset grid, but with the same sampling rate, as shown in figure 1.

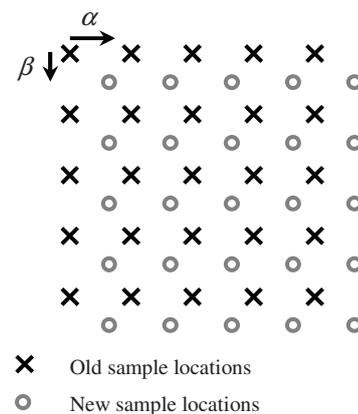


Figure 1 Input and output sample grids.

This resampling can be implemented in a separable fashion, where each row of the image is treated as an independent 1D signal and resampled using a 1D resampling filter  $h[m]$ . Then the same procedure is carried out over the columns. Considering this 1D operation on a single row of an image:

$$\hat{f}_\alpha[k] = \sum_{m \in \mathbb{Z}} f[k+m]h[m], \quad (1)$$

where  $f[k]$  is the 1D signal that is being resampled and  $\hat{f}_\alpha[k]$  is the output. The global, space-invariant least-squares optimal kernel can be computed using:

$$h = \arg \min_h \sum_k \left( f_\alpha[k] - \sum_{m \in \mathbb{Z}} f[k+m]h[m] \right)^2 \quad (2)$$

where  $f_\alpha[k]$  is the image  $f$  sampled on the grid shifted by  $\alpha$  (i.e. the ground-truth that the ideal resampled image should match). This minimization must be performed subject to the ‘‘partition of unity’’ constraint:

$$\sum_{m \in \mathbb{Z}} h[m] = 1 \quad (3)$$

to make sure that there is no overall gain in flat regions of the image [1, 13].

### B. Model-based Near-optimal Interpolation

In previous work [9] we have shown that optimizing a kernel on one image and then using it to resample a different image yields reasonable results. Later, we described how to optimize the resampling kernel on an analytic image model [10]. Such a kernel should be near-optimal for a class of images that the model represents. One such proposed model was based on the fact that, to a first approximation, most images are piece-wise constant, with relatively flat areas separated by step-edges. This model resulted in resampling kernels visibly similar to optimal kernels. Experimental tests showed that the performance of these model-based kernels was very close to the performance of the optimal kernels. Here, we reuse this model and demonstrate how a 1D kernel for image interpolation in a separable fashion can be derived using least-squares optimization. The derivations are demonstrated using a fourth order resampling filter, but filters of other orders can be derived in a similar manner.

We employ a 1D step-edge model:

$$P(x; p) = \begin{cases} 1 & x \leq p \\ 0 & x > p \end{cases}, \quad (4)$$

where  $p$  is the location of the step-edge, and derive the resampling kernel in the following way.

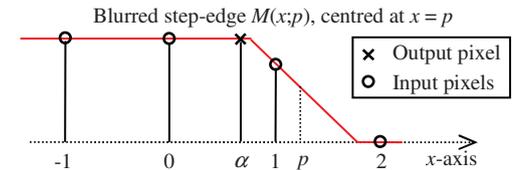
The fourth order filter has a region of support of five pixels (this region always includes four input pixels; hence fourth order). Consider interpolating an image consisting of many step-edges. The filter window will be positioned over every output sample and a step-edge may (or may not) fall somewhere within the filter window, assuming that the step-edges are separated by at least five pixels so that only one step-edge may fall within the filter window. If the step-edges do not exhibit a periodic pattern with a period that is a rational multiple of the sampling period, the step-edges will fall at random positions within the filter window, positioned around different output pixels.

To simplify the derivation, rather than considering many such filter windows, consider a single window with input pixels located at  $-1, 0, 1$  and  $2$ . Instead of centring the filter window on the output pixel, let the output pixel position be between  $0$  and  $1$ . This keeps the filter window static and considerably simplifies the derivation of the contribution of each input pixel within the filter window as a function of the location of the output pixel.

The input pixels are from an area-sampled (area-sampling is used to be consistent with the imaging process) step-edge that is located somewhere within the region of support, but at an unknown position. To make things easier, area-sampling a step-edge can be replaced by point-sampling the step-edge convolved with a rectangular pulse with height one and width of one pixel:

$$M(x; p) = \begin{cases} 1 & x \leq p - \frac{1}{2} \\ p + \frac{1}{2} - x & p - \frac{1}{2} < x \leq p + \frac{1}{2} \\ 0 & p + \frac{1}{2} < x \end{cases} \quad (5)$$

Figure 2 demonstrates an example of this process, where the output pixel is located at  $\alpha = 0.63$  and the edge is located at  $p = 1.2$ .



$$\begin{array}{lll} M(-1; 1.2) = 1 & M(0.63; 1.2) = 1 & M(2; 1.2) = 0 \\ M(0; 1.2) = 1 & M(1; 1.2) = 0.7 & \end{array}$$

Figure 2 Input and output pixels within the filter window sampling a blurred step-edge located at  $p=1.2$ .

Analogous to (1), the interpolated value at  $x = \alpha$  is computed using a weighted sum of the input pixels. The square error between the actual output and the interpolated value can be expressed as

$$\epsilon^2 = \left( M(\alpha; p) - \sum_{m=-1}^2 M(m; p)h[m] \right)^2 \quad (6)$$

If a step-edge can fall anywhere within the filter window with equal probability, then it is desirable to optimize the resampling filter for any such case, which can be accomplished by minimizing the square error over all possible locations of the step-edge within the filter window with respect to the filter coefficients:

$$h = \arg \min_h \int_{-1.5}^{2.5} \left( M(\alpha; p) - \sum_{m=-1}^2 M(m; p) h[m] \right)^2 dp \quad (7)$$

This minimization must be performed subject to the “partition of unity” constraint, stated by (3), which can be built directly into the optimization by substituting

$$h[0] = 1 - h[-1] - h[1] - h[2] \quad (8)$$

into (7). After some rearranging, this gives

$$h = \arg \min_h \int_{-1.5}^{2.5} \left( \bar{M}(\alpha; p) - \sum_{m=-1,1,2} \bar{M}(m; p) h[m] \right)^2 dp, \quad (9)$$

where  $\bar{M}(x; p) = M(x; p) - M(0; p)$ . This is a standard linear least-squares problem – to solve it, the objective function in (9) is differentiated with respect to each filter coefficient, and the partial derivatives are equated to zero. This results in a set of three linear equations, best represented in matrix form:

$$\mathbf{A}\mathbf{h} = \mathbf{b}, \quad (10)$$

where  $\mathbf{A}$  is the following  $3 \times 3$  matrix

$$\mathbf{A} = \begin{bmatrix} \int \bar{M}_{-1} \bar{M}_{-1} dp & \int \bar{M}_{-1} \bar{M}_1 dp & \int \bar{M}_{-1} \bar{M}_2 dp \\ \int \bar{M}_1 \bar{M}_{-1} dp & \int \bar{M}_1 \bar{M}_1 dp & \int \bar{M}_1 \bar{M}_2 dp \\ \int \bar{M}_2 \bar{M}_{-1} dp & \int \bar{M}_2 \bar{M}_1 dp & \int \bar{M}_2 \bar{M}_2 dp \end{bmatrix}, \quad (11)$$

where all integrals have the same limits of -1.5 and 2.5, and  $\mathbf{h}$  and  $\mathbf{b}$  are  $3 \times 1$  vectors defined as

$$\mathbf{h} = \begin{bmatrix} h[-1] \\ h[1] \\ h[2] \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \int \bar{M}_{-1} \bar{M}_\alpha dp \\ \int \bar{M}_1 \bar{M}_\alpha dp \\ \int \bar{M}_2 \bar{M}_\alpha dp \end{bmatrix} \quad (12)$$

where  $h_m$  denotes  $h[m]$  and  $\bar{M}_m$  denotes  $\bar{M}(m; p)$ . The solution to (10) can be simply expressed as

$$\mathbf{h} = \mathbf{A}^{-1} \mathbf{b} \quad (13)$$

All elements of  $\mathbf{A}$  and  $\mathbf{b}$  have the same structure and we examine the first element of  $\mathbf{A}$  as an example:

$$\int_{-1.5}^{2.5} \bar{M}(-1; p)^2 dp = \int_{-1.5}^{2.5} (M(-1; p) - M(0; p))^2 dp \quad (14)$$

Computing this integral is relatively trivial and is depicted in figure 3 in graphical form. The other elements can be computed in exactly the same way and then the filter weights can be computed by evaluating (13). Recall that these weights are functions of the location of the output pixel  $\alpha$ , which is defined to be between 0 and 1. To compare it to other interpolation functions, we can express this in the form of an interpolation kernel (by substituting  $-(\alpha+1)$  for  $\alpha$  into  $h[-1]$ ,  $-(\alpha-1)$  into  $h[1]$ ,  $-\alpha$  into  $h[0]$  and  $-(\alpha-2)$  into  $h[2]$  and using the appropriate piece of each function):

$$\varphi_4(x) = \begin{cases} \frac{64}{56}|x|^3 - \frac{117}{56}|x|^2 - \frac{3}{56}|x| + 1 & 0 \leq |x| \leq 1 \\ -\frac{24}{56}|x|^3 + \frac{129}{56}|x|^2 - \frac{219}{56}|x| + \frac{114}{56} & 1 \leq |x| \leq 2 \\ 0 & 2 \leq |x| \end{cases} \quad (15)$$

The resulting interpolation kernel is a piece-wise cubic polynomial, similar to what is commonly known as Keys’ cubic convolution [14]:

$$\varphi_{cc}(x) = \begin{cases} \frac{3}{2}|x|^3 - \frac{5}{2}|x|^2 + 1 & 0 \leq |x| \leq 1 \\ -\frac{1}{2}|x|^3 + \frac{5}{2}|x|^2 - 4|x| + 2 & 1 \leq |x| \leq 2 \\ 0 & 2 \leq |x| \end{cases} \quad (16)$$

Keys’ cubic convolution is of special interest as it tends to give more accurate results than other piece-wise cubic kernels [15]. Both of these kernels are plotted in figure 4. It can be observed that the kernels have only a subtle difference in shape, even though they were derived using two very different methods. However, unlike Keys’ polynomial, our kernel is not continuously differentiable (compare figure 5 and figure 6). This would have a consequence if used to resample smooth functions; however, images generally contain many discontinuities. The discontinuity in the first (and following) derivatives starts to slowly disappear as the filter order is increased, apart from near the outside of the support region. An interesting observation is that as the order of the optimal kernel is increased, it begins to approximate the cubic spline. A large portion of the kernel around the centre of support is identical to the cubic spline; however, near the edges of the support region the two functions deviate and our kernel is not continuously differentiable.

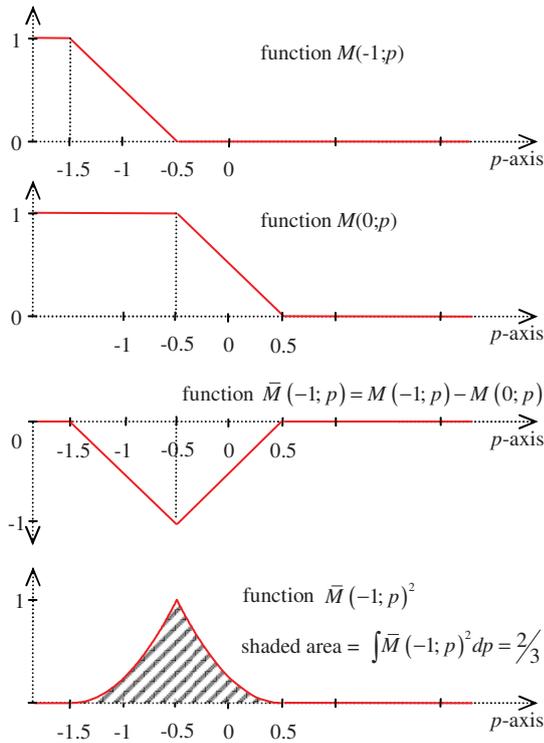


Figure 3 Graphical demonstration of the solution of (14). First  $M(0;p)$  is subtracted from  $M(-1;p)$ , then this difference function is squared and finally integrated to give the area under the curve. Notice that the integrated function has limited spatial support. This is also true for all other elements of **A** and **b**, so integration limits can be set to infinity.

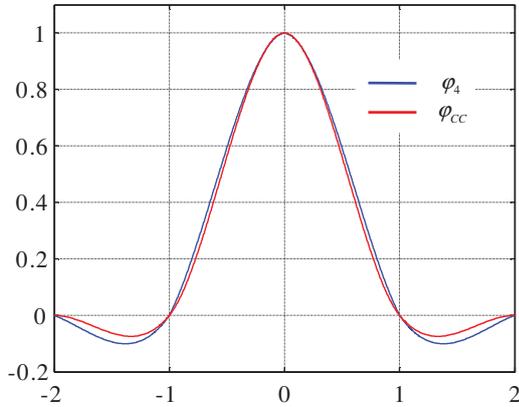


Figure 4 The proposed kernel and Keys' cubic convolution kernel.

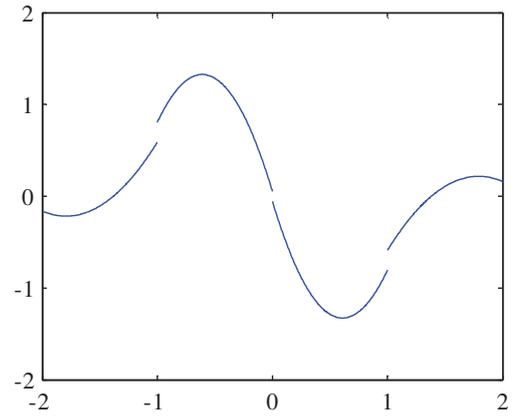


Figure 5 First derivative of the proposed kernel.

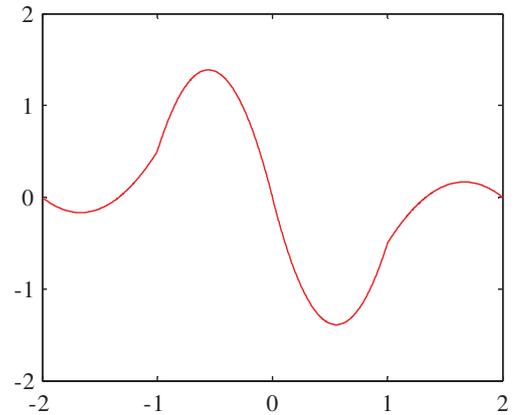


Figure 6 First derivative of Keys' cubic convolution.

### III. QUANTITATIVE EVALUATION

We compare the proposed resampling kernels of order four and six (obtained in the same way as order four) to a number of existing kernels with similar extent of spatial support: Keys' cubic convolution, Lanczos kernel [16] and cubic spline [17]. We also compare with the results of a model-based 2D non-separable kernel [11].

#### A. Experimental set-up

We evaluate the methods by resampling a simulated test image on a sampling grid shifted by half a pixel in each direction and computing root-mean-square-error (RMSE) between the resampled image and simulated ground-truth output.

The input and output images are created from a higher resolution (2592×3888 pixels) image using an area-sampling imaging model. The high-resolution image is blurred using a four by four rectangular filter with flat response to simulate area-sampling and sub-sampled by a factor of four to create the input and the desired output images (the high-resolution image is shifted by two pixels in each direction before sub-sampling

to create the output image) followed by quantisation to 256 grayscale levels.

The experiment is conducted with images of four different scenes pictures in figure 7. These scenes provide a variety of compositions of high-frequency detail such as step-edges and texture and relatively low-frequency flat areas and should indicate how well the proposed kernel can resample different types of scenes.



New Zealand is a constitutional monarchy with a parliamentary democracy. Although it has no written constitution, the Constitution Act 1986 is the principal formal statement of New Zealand's constitutional structure. Queen Elizabeth II is the head of state and is titled Queen of New Zealand under the Royal Titles Act (1953). She is represented by the Governor-General, currently Anand Satyanand.

The Governor-General exercises the Crown's prerogative powers, such as the power to appoint and dismiss ministers and to dissolve Parliament, and reserve powers. The Governor-General also chairs the Executive Council, which is a formal committee consisting of all ministers of the Crown. Members of the Executive Council are required to be Members of Parliament, and most are also in Cabinet. Cabinet is the most senior policy-making body and is led by the Prime Minister, who is also, by convention, the Parliamentarian.

Figure 7 Test images 'bird', 'cat', 'face' and 'text'.

The Lanczos kernel is defined as

$$\varphi_{\text{Lanczos}}(x) = \begin{cases} \text{sinc}(x)\text{sinc}\left(\frac{x}{w}\right) & 0 \leq |x| \leq w \\ 0 & w \leq |x| \end{cases} \quad (17)$$

We employ fourth ( $w = 2$ ) and sixth ( $w = 3$ ) order filters for comparison to our proposed model-based kernels. The cubic spline interpolation is employed as implemented by the *interp2* command in Matlab 2008a [18]. For a detailed derivation of the 2D non-separable kernel, please refer to [7].

## B. Results

The results of this experiment are displayed in figure 8 in the form of a bar graph with separate axes for each test scene. The percentage number on each bar indicates improvement in RMSE in comparison to Keys' cubic convolution.

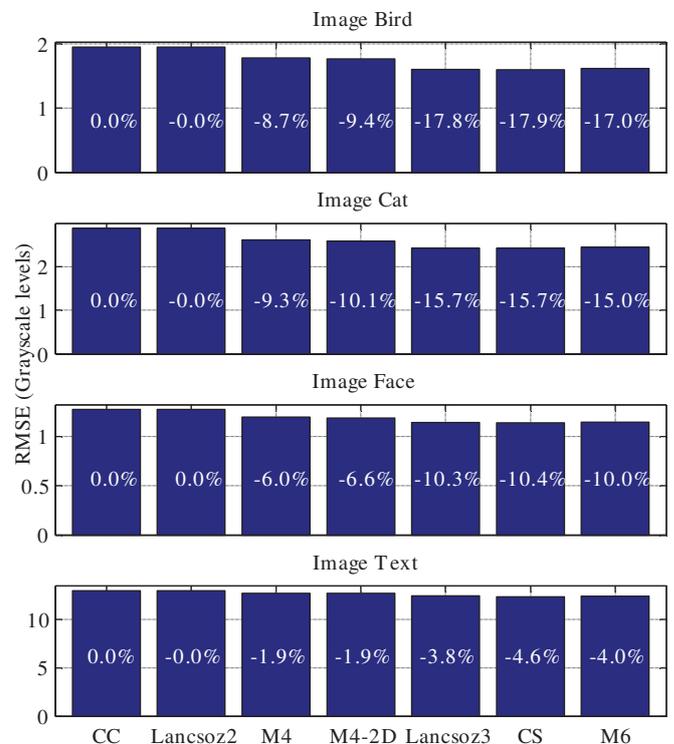


Figure 8 RMSE in grayscale levels resulting from resampling images 'bird', 'cat', 'face' and 'text' using the following methods: cubic convolution (CC), Lanczos2, proposed model-based order 4 (M4), 2D non-separable model based 4x4 (M4-2D), Lanczos3, cubic spline (CS) and proposed model-based order 6 (M6). Percentage number indicates improvement from cubic convolution.

Keys' cubic convolution results in very similar performance as the Lanczos kernel with the same spatial support. The fourth order model-based kernel performs better than both of these for all four test images; however, the actual improvement varies from image to image. As all of these have the same order, they are also of the same computational complexity and our proposed kernel gives smaller error for the same complexity.

The non-separable model-based kernel performs slightly better than the separable kernel. This improvement comes at a significant computational cost. It would actually make more sense to use a higher order separable kernel as these get better performance than the 4×4 non-separable kernel at a smaller increase in computational complexity.

The cubic spline, with its much large spatial support consistently results in the smallest error; however, the Lanczos3 filter and the proposed filter, both of order six result in almost the same improvement.

#### IV. CONCLUSIONS

Image interpolation is required whenever digital images need to be resampled on a different sampling grid, due to their inherent discreteness. To reduce computational costs this is generally performed in a separable manner using 1D resampling kernels. In this work we have proposed a novel resampling kernel based on least squares optimization of a piece-wise constant image model. The derivation was demonstrated using a fourth order resampling filter, but is easily extendable to any order. The resulting fourth order kernel is a piece-wise cubic polynomial similar to Keys' cubic convolution. However, the kernels derived with much wider support region begin to approximate the cubic spline.

We experimentally compared the performance of the proposed resampling kernels of orders four and six to other resampling filters of order four (cubic convolution, Lanczos2) and order six (Lanczos3) and also the cubic spline. The results indicate that lower RMSE can be achieved using our step-edge based resampling filter of order four than other filters of the same order. The proposed kernel of order six performed similarly to the Lanczos3 kernel and the cubic spline.

These initial results show that improved resampling can be achieved without any additional computational costs; however more testing and analysis of the proposed kernel is required.

#### REFERENCES

- [1] M. Unser, "Sampling-50 years after Shannon," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569-587, 2000.
- [2] E. H. W. Meijering, W. J. Niessen, and M. A. Viergever, "The sinc-approximating kernels of classical polynomial interpolation." pp. 652-656 vol.3.
- [3] T. M. Lehmann, C. Gonner, and K. Spitzer, "Survey: Interpolation methods in medical image processing," *IEEE Transactions on Medical Imaging*, vol. 18, no. 11, pp. 1049-1075, Nov, 1999.
- [4] T. M. Lehmann, C. Gonner, and K. Spitzer, "B-spline interpolation in medical image processing," *IEEE Transactions on Medical Imaging*, vol. 20, no. 7, pp. 660-665, Jul, 2001.
- [5] S. E. Reichenbach, and F. Geng, "Two-dimensional cubic convolution," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 857-865, Aug, 2003.
- [6] J. Z. Shi, and S. E. Reichenbach, "Image interpolation by two-dimensional parametric cubic convolution," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 1857-1870, Jul, 2006.
- [7] A. Gilman, "Least-squares optimal interpolation for direct image super-resolution," Ph.D. dissertation, Massey University, Palmerston North, New Zealand, 2009.
- [8] R. R. Schultz, and R. L. Stevenson, "A Bayesian approach to image expansion for improved definition," *IEEE Transactions on Image Processing*, vol. 3, no. 3, pp. 233-242, May, 1994.
- [9] A. Gilman, and D. G. Bailey, "Near optimal non-uniform interpolation for image super-resolution from multiple images," in *Image and Vision Computing New Zealand*, Great Barrier Island, New Zealand, 2006, pp. 31-35.
- [10] A. Gilman, D. G. Bailey, and S. R. Marsland, "Interpolation models for image super-resolution," in *4<sup>th</sup> IEEE International Symposium on Electronic Design, Test and Applications*, Hong Kong, 2008, pp. 55-60.
- [11] A. Gilman, D. G. Bailey, and S. R. Marsland, "Least-squares optimal interpolation for fast image super-resolution," unpublished.
- [12] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: A technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21-36, May, 2003.
- [13] P. Thévenaz, T. Blu, and M. Unser, "Image interpolation and resampling," *Handbook of medical imaging*, pp. 393-420, Orlando, FL, USA: Academic Press, Inc., 2000.
- [14] R. G. Keys, "Cubic convolution interpolation for digital image-processing," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 29, no. 6, pp. 1153-1160, 1981.
- [15] E. Meijering, and M. Unser, "A note on cubic convolution interpolation," *IEEE Transactions on Image Processing*, vol. 12, no. 4, pp. 477-479, Apr, 2003.
- [16] K. Turkowski, "Filters for common resampling tasks," *Graphics gems*, pp. 147-165: Academic Press Professional, Inc., 1990.
- [17] M. Unser, "Splines - a perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22-38, Nov, 1999.
- [18] MathWorks. "Interp2," 28 September, 2009; <http://www.mathworks.com/access/helpdesk/help/techdoc/ref/interp2.html>.