Interpolation Models for Image Super-resolution

Andrew Gilman, Donald G. Bailey, Stephen R. Marsland
Institute of Information Sciences and Technology
Massey University, Palmerston North, New Zealand
agilman@iee.org, d.g.bailey@massey.ac.nz, s.r.marsland@massey.ac.nz

Abstract

Image super-resolution involves interpolating a non-uniformly sampled composite image at uniform locations of a high-resolution image. Interpolation methods used in the literature are generally based on arbitrary functions. Optimal (in least squares sense) interpolation kernels can be derived if the ground-truth high-resolution data is known, this is obviously impractical. An observation that the optimal kernels for very different images are similar suggests that a kernel derived on one image can interpolate another image with good results. This paper extends this idea by developing two image models that capture the important characteristics of an image and uses the models to derive optimal kernels. One of the models results in linear interpolation and the other one results in a piece-wise cubic kernel similar to that of cubic spline. This later model is experimentally shown to be near optimal for three different images. The notion of deriving optimal interpolators from the image model and the model of image capturing process provides a unifying framework that brings together linear and cubic interpolators and gives them a theoretic backing.

Keywords: optimal, interpolation, super-resolution, non-uniform, least-squares, image, model

1 INTRODUCTION

Digital image capture produces discrete representations of continuous scenes. This discretisation in both space and intensity is a sampling process that creates aliasing, and information at frequencies above the Nyquist rate is lost. It is common to wish to construct a higher resolution image from a template image or a set of images, but the aliasing and loss of frequency information makes this an ill-posed (inverse) problem.

The typical solution to this problem (known in the literature as image super-resolution reconstruction, or simply super-resolution) is to use an ensemble of related lower-resolution images. As each of these images has aliased the higher frequency information slightly differently, under certain conditions it is possible to 'unwrap' some of the aliasing and reconstruct the lost higher frequencies.

This paper describes a novel method of interpolation from irregularly spaced data points that is suitable for super-resolution. The method is demonstrated on the problem of super-resolution of an image based a number of lower-resolution images that capture scene from a slightly different positions. In this work, only a global translational motion is considered, created, for example, by a camera shifted laterally by a small amount. Images within the ensemble can be assumed to have a minute displacement relative to each other. Ideally, the relative displacement should be subpixel in order to increase the overlap between all of the images and to minimise the change in perspective. While the methods are derived for one dimensional images for space reasons, the extension to two dimensions is not difficult and follows the same form.

There are numerous methods [1] of performing super-resolution. Many of them are computationally expensive in nature, but allow for complicated motion models, significant noise and image degradation, and other aspects that are not considered in this work. Given assumptions of global translational motion, low noise and linear space and time invariant blur due to the imaging sensor point spread function (PSF), image super-resolution reconstruction can be split into three distinct steps:

- Registration
- Reconstruction \ Interpolation
- Deblurring

Image registration is a technique that can be used to determine the relative translations between the input images. Generally, the desire is to do this from the contents of the images alone, without any prior knowledge. There are many different methods for performing registration [2]; however, in the context of image super-resolution, image registration is required to determine the offsets between the images with accuracy down to a small fraction of a pixel [3].

Once the images have been registered, all the pixels from the ensemble can be combined to form a composite image. The resultant image is no longer sampled on a uniform rectangular grid, but due to global translational motion, it has a semi-uniform structure, as can be seen in Figure 1. Reconstructing the image data at all points on a high resolution grid requires that the semi-regular data is interpolated and resampled. It is this interpolation problem that is addressed in this paper.

0-7695-3110-5/08 $25.00 © 2008 IEEE
DOI 10.1109/DELTA.2008.104

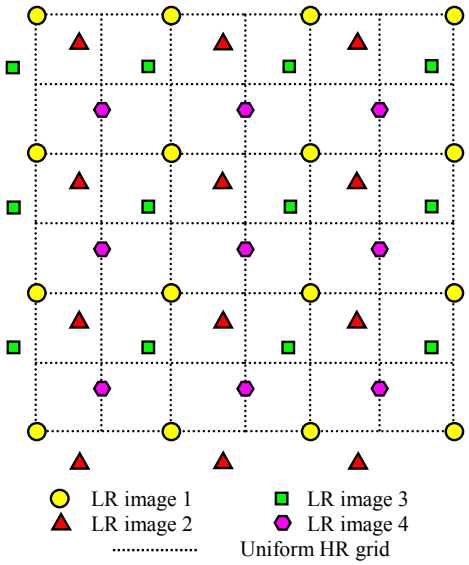

Figure 1. Composite image exibits a semi-uniform structure.

For the full super-resolution approach, a deblurring procedure can now be applied that restores the high frequencies that have been suppressed by the low-resolution imaging process. In this paper we do not perform this deblurring, as it obfuscates the results of the interpolation and makes ground truth comparisons of different interpolation methods impossible. For this reason, the results of experiments are reported using comparison with ground truth values from high resolution images blurred by the PSF from the low-resolution imaging process.

2 INTERPOLATION

Digital images are samples of a continuous scene and are inherently defined on a discrete set of locations (generally a regular rectangular grid, but can also be irregular). However, it is often desired to obtain the value of a point at an arbitrary location. This requires reconstructing the underlying continuous function, which can then be resampled at arbitrary locations.

This problem is referred to as interpolation. Unfortunately, unless there is some prior knowledge about the shape of the underlying function, this problem is under-determined – there is an infinite number of continuous functions that pass through the defined sample points. Different interpolation methods make different assumptions about the underlying continuous function in order to select a unique solution (e.g. band-limitedness, smoothness, etc).

Given a continuous function representing the captured scene (blurred by the PSF) $g(x)$, $x \in \mathbb{R}$, let us define a discrete input image $f_I(p)$ sampled at a set of discrete grid locations P, such that:

$$f_I(p) = \begin{cases} g(p) & \text{on the grid} \\ \textit{undefined} & \text{elsewhere} \end{cases} \quad (1)$$

The problem of interpolation is then, to estimate the continuous image $\tilde{g}(x)$ from $f_I(p)$.

If we consider interpolation to be a filtering function, this can be implemented as a spatial convolution. Then, the reconstructed continuous function is a linear combination of input values:

$$\tilde{g}(x) = \sum_p f_I(p) H_p(x-p), \quad p \in P, \quad (2)$$

where $H_p(\cdot)$ is some interpolation function associated with input point p.

In the case of resampling a digital image it is not necessary to reconstruct the complete continuous surface; only the values at the new sample locations are required. Let $f_O(q)$ be the output discrete image, sampled at new locations $q \in Q$:

$$\begin{aligned} f_O(q) &= \tilde{g}(q) \\ &= \sum_p f_I(p) H_p(q-p), \quad q \in Q,\ p \in P \end{aligned} \quad (3)$$

Since images in general have limited spatial correlation, it makes little sense to use all of the input samples to interpolate each desired point. Generally, only a small neighbourhood of input points that are within a certain distance to the interpolated point are used:

$$f_O(q) = \sum_p f_I(p) H_p(q-p), \quad q \in Q,\ p \in P_{q,w}, \quad (4)$$

where $P_{q,w} \triangleq \left\{ p \mid p \in P, \|p-q\|^2 < w \right\}$ is a set of points within some predetermined distance w of the output point q (also called the region of support).

Standard methods in literature include the spline approximations (typically cubic splines in one dimension, and thin-plate and B-splines in two dimensions) as well as linear approximations. However, the choice of interpolant is arbitrary and does not generally imply an understanding of the underlying data or image capturing process. While experimentation can show that one method works better than another for some particular dataset, in this paper we suggest an alternative approach where the interpolant is optimised (in the least squares sense) on a typical image model.

2.1 Optimal Interpolation

Instead of using an arbitrary interpolation function, we can derive the interpolator that is optimal for a particular image, the input locations P and desired sample locations Q. For image processing it is common to measure reconstruction error as the mean-squared error (MSE). This corresponds to minimising

the L_2 norm $\|g(q)-f_O(q)\|^2$, which leads to an unbiased estimator. Another benefit is that it (obviously) leads to a least-squares optimisation problem, which can be solved directly rather than iteratively for this case.

For super-resolution problem there is some regularity in the semi-uniform composite image (see Figure 1, where the same pattern of points is repeated throughout the image). The method proceeds by exploiting this regularity, by reasoning that where the distribution of data points within the region of support of a particular output pixel is the same, the weights allocated to the input pixels inside that support is also the same in the case of all of those output pixels. This set of weights can then be optimised to minimise common error amongst all of those output pixels.

In 1D, if the resolution is to be increased by an integer factor N then there will be N of these local neighbourhood patterns and hence N sets of weights to be individually optimised.

Let us break all of the output points Q into N sets Q_n, such that the regions of support of the points in each set have a common pattern. This is demonstrated in Figure 2, where a compound image is resampled at uniform locations with double the sampling rate of input images. It consists of two low resolution images with offsets of 0.0 and 0.2 relative to the high resolution grid. The interpolated grid locations are indicated with dashed vertical lines and regions of support with $w = 0.5$ are shown for each interpolated point. It can be seen that all region of support patterns for points in Q_1 are identical in terms of relative positions of the input and output pixels and the same for set Q_2.

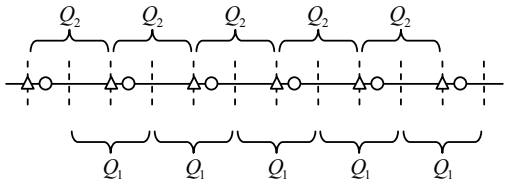

Figure 2. Regions of support of two filters required to resample a 1D compound image.

For each pattern Q_n, the least-squares optimisation that is required can be formulated as:

$$\min_H \sum_{Q_n} \left\| g(q) - \sum_p f_I(p) H_p(q-p) \right\|^2, \quad (5)$$

where $q \in Q_n$, $p \in P_{q,w}$ and subject to the constraint:

$$\sum_p H_p(q-p) = 1 \;\forall q \quad (6)$$

Obviously, optimal interpolation is impractical because it requires knowing the exact output to calculate an approximation of the output! However, optimal interpolation kernels derived from ground truth (simulated) data can provide useful insights into the interpolation process.

2.2 Simulated Optimal Interpolation

In this section we derive optimal interpolation kernels for three distinct images, pictured in Figure 3. Image 'bird' offers a typical scene with many occluding objects, resulting in many sharp edges and relatively flat areas in between. Image 'cat' includes many highly textured areas, which contain a lot of high-frequency information. And image 'face' is a typical image which contains a combination of both texture (e.g. hair) and edges. The subject of the image – a human face – is also very familiar to us, which is good for assessing reconstruction error qualitatively.

The optimal kernels for a particular interpolation depend not only on the input data but also the relative position of input samples. Here we consider two example situations; both reconstruct the output from two input images. In the first case the input images have shifts relative to the output grid of 0.0 and 0.5 low resolution pixels (i.e. the input samples are uniformly spaced); and the second case the shifts are 0.0 and 0.2 of a low resolution pixel.

The discrete interpolation kernels are hard to visualise and compare because they also depend on the relative location of the output samples. Therefore it is more meaningful to compare the continuous interpolation functions, $H_p(x-p)$ which specify the contribution of input pixel located at p to reconstructing the continuous image at location x. Although in general each input point p would have its own interpolation function $H_p(x-p)$, in the case of a global translational motion all input pixels belonging to a single input image have the same interpolation

Figure 3. Test images: bird, cat, and face (Size reduced for display purposes)

function (which is replicated at each of those pixel's positions).

Unfortunately, deriving the continuous interpolation functions is not possible because we do not know the continuous underlying functions of the three images. However, the dimensions of the source images are 2592x3840 pixels; hence, if the input low-resolution images are much smaller than this, the optimal kernels can be derived at a relatively high sample rate.

To derive 1D kernels the image data is treated in one dimension. To generate a low-resolution image from the source image, it is filtered using a 1x40 horizontal box average filter to simulate 1D area sampling. Then the image can be shifted by the required integer offset (for example 20 samples for offset of 0.5 and 8 samples for offset of 0.2) and downsampled by a factor of 40 horizontally.

Then, by reconstructing the source image (blurred by the low-resolution PSF) at full resolution, we derive the interpolation functions at locations spaced by $1/40^{th}$ of a pixel. For these derivations we have used region of support $w = 3$, with the resulting kernels plotted in Figure 4. Kernels for only one of the input images are shown in each case – by symmetry, the kernels for the second input image are a mirror image of these.

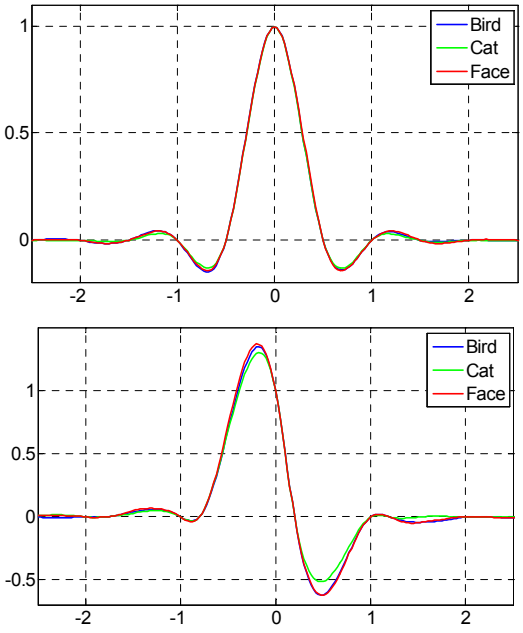

Figure 4. Optimal interpolation functions derived from images 'Bird', 'Cat', and 'Face' at sample rate of $1/40^{th}$ of a pixel and linearly interpolated for display purposes.

It can be clearly seen that the kernels generated from all three images are very similar. We only show kernels for two sets of relative displacements, but this generalises to other combinations of input image shifts.

From this we can conclude that the optimal kernels are only weakly dependent on the image content and using a kernel optimised on one image to interpolate another will result in a small increase in MSE from optimal. This supports the findings of reference [4], where it was shown that optimising a 2D kernel on one image and then using it to interpolate another image results in near optimal MSE for many different combinations of pixel positions.

2.3 Optimal Interpolation Using a Model

Since the optimal kernels are almost the same for very different images, this implies that the dominant factor is the relative offset between the low resolution samples. There is some dependence on the actual image content, but if we a model that has similar critical characteristics to a class of images, then the optimal kernels derived from that model should be near optimal for that image class. For a simple model, a closed form expression can be found for the weights as a function of the input sample positions, significantly speeding up the process.

Images, or parts of images that have a lot of high-frequency information, as for example the textured areas in image 'cat' or the hair in image 'face' would benefit from a model that has a wide frequency extent. A simple model comprised of a set of randomly located impulses should have similar characteristics to such images.

Most images comprise of multiple occluding objects. This leads to many sharp edges that separate these objects. Frequently the areas between the edges are relatively flat compared to the height of the edges. Image 'bird' is an extreme example of this – to a first approximation it can be considered to be piece-wise constant with sharp edges separating the constant regions. This suggests that a set of randomly located step edges would be a suitable model to capture the important characteristics of such images.

With these two models, impulses and step edges, when considering a single region of support, we can consider an equivalent model of a single impulse or single step edge randomly located within the region of support. Since we cannot predict where the impulse or edge is located with respect to either the input or output sampling grids, the optimal kernels for a particular model are determined in such a way as to minimise the expected error over all possible locations.

To be consistent with the imaging model, the impulse or edge is convolved with a rectangular pulse (with width equal to one low-resolution pixel) to simulate area sampling. The impulse becomes a rectangular pulse and the step edge becomes a linear ramp, both of width equal to one low-resolution pixel:

$$M_1(x,\alpha) = \begin{cases} 1 & \alpha - 0.5 \le x < \alpha + 0.5 \\ 0 & \text{otherwise} \end{cases} \qquad (7)$$

$$M_E(x,\alpha)=\begin{cases}1 & x<\alpha-0.5\\ \alpha+0.5-x & \alpha-0.5\le x<\alpha+0.5\\ 0 & x>\alpha+0.5\end{cases} \quad (8)$$

where α is the location of the centre of the blurred impulse or edge relative to the centre of the region of support ($x=0$).

Let us define a region of support P_n around the output point, $x=0$, with a common pattern of input pixel locations shared with a set of output points Q_n :

$$P_n \triangleq P_{q,w}-q, \quad \forall w,q\in Q_n \quad (9)$$

Then the desired output is $M(0,\alpha)$ with the input set defined as

$$f_I(p,\alpha)=M(p,\alpha), \quad p\in P_n \quad (10)$$

Assuming that α is a random variable with continuous uniform probability distribution, equation (5) becomes

$$\min_H \int_{-w-\frac{1}{2}}^{w+\frac{1}{2}} \left\| M(0,\alpha)-\sum_p f_I(p,\alpha)H_p(-p)\right\|^2 d\alpha, \quad (11)$$

where $p\in P_n$.

To solve (11), the function is differentiated with respect to the weights $H_p(-p)$ and equated to zero to give a system of equations linear in $H_p(-p)$, which can be solved by any standard method.

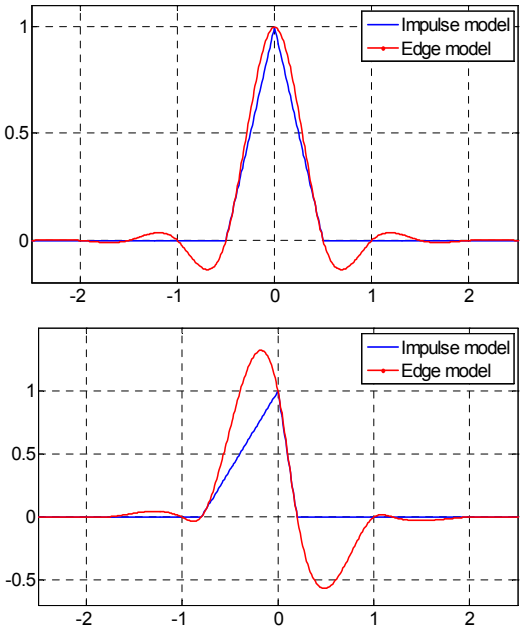

Figure 5. Optimal kernels generated from impulse and edge models for input image offsets of [0 0.5] and [0 0.2].

Figure 5 shows the resulting kernels from the two models. There are again two input images located at 0 and 0.5 low-resolution pixels in the first plot and 0 and 0.2 in the second plot.

It can be seen that the impulse model only uses two nearest input points only (one on the left and one on the right of the output pixel) and performs linear interpolation between those. This is independent of the size of the region of support, as long as the region of support included at least one input sample on each side of the output sample.

The optimal kernels for edge model are piecewise cubic with separate cubic functions between the input sample locations. These kernels resemble closely the optimal kernels shown in Figure 4, implying that edge model captures the important characteristics of these images.

3 RESULTS

In this section we assess the performance of the kernels produced using the models described in previous section. The procedure is similar to the one described in [4]. We use two low-resolution images to reconstruct an image with twice the sampling rate of the input.

With smallest possible offset of 1/40th of a pixel and two input images, there are 779 possible combinations of unique input image displacements. Instead of performing a Monte Carlo simulation using randomly selected combinations of displacements (as in [4]), all of the possible combinations are used. The errors are analysed in a similar way, with MSE calculated for each combination and sorted in ascending order to give the inverse cumulative distribution function (iCDF).

The effectiveness of the models is determined by comparing the results with other standard methods. Since the edge model gives a piecewise cubic kernel, suitable methods for comparison are the cubic spline, piece-wise cubic Hermite interpolating polynomial (PCHIP) [5], and modified Shepard's method [6]. These methods are used extensively for interpolation, and are known to give good results despite the fact that they are arbitrary.

The cubic spline and PCHIP interpolation are used as implemented by the Matlab 7.1 *interp1* function [5]. Our implementation of the modified Shepard's method uses least squares to fit a cubic polynomial to the points within the region of support, where each point's value is weighted by its inverse distance to the output point.

The iCDF results for one of the images ('cat') are plotted in Figure 6, with median MSE tabulated in Table 1 for the other images. We choose to report the median because it is a robust estimator of the performance.

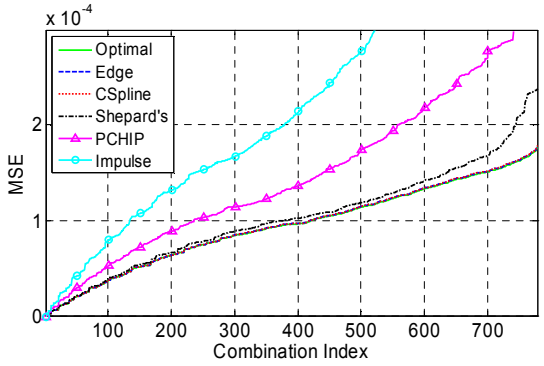

Figure 6. Inverse cumulative distribution of the resultant MSE for different methods.

Table 1: Median MSE for the images relative to optimal interpolation.

	bird	**cat**	**face**
Optimal	1.58E-04 100%	9.69E-05 100%	2.36E-05 100%
Edge Model	100.5%	100.5%	101.7%
Cubic Spline	100.5%	100.5%	101.7%
Shepard's	113.1%	105.4%	112.2%
PCHIP	178.7%	138.8%	151.8%
Impulse Model	295.7%	215.2%	299.9%

For uniform sampling, the interpolating kernels from the edge model are identical to the cubic spline kernels for a wide region of support ($w \geq 2$). However, for non-uniform sampling the kernels become slightly different as the offset between the input images becomes smaller. For narrow regions of support, the edge model kernels are only C^0 whereas the smoothness constraint of the spline makes it twice differentiable at the knot points. The difference in MSE is negligible (less than 0.1%).

Even though the impulse model does not perform as well as the edge model, it does perform significantly better on the 'cat' image than the other two. This supports the hypothesis that the contents of 'cat' is more impulse-like than the other two images. However, on closer examination, it was found that even the fine detail (such as the whiskers and fur) are still several very high resolution pixels wide. As a consequence, the edge model out-performed the impulse model.

4 Conclusions

Optimal interpolation kernels have been derived for a set of images, simulating the image capture process by using a very high resolution "ground truth" image. For a range of quite different images, the optimal kernels are very similar. The fact that they are only weakly dependent on image content implies that kernels derived from one image should give near optimal results on other images, particularly images with similar characteristics.

This result led to the proposal of using an analytic model of the image to capture the important characteristics of typical images. Deriving kernels from these models avoids the need for having ground truth data for performing the optimisation.

Two simple models based on observations of the image characteristics have been proposed. The optimal kernels for an impulse model are simple to derive analytically, and result in standard linear interpolation between the nearest sample on each side of the output sample. A step edge model considers images to be approximately piecewise constant with step edges between the regions. This results in piecewise cubic kernels that are very similar to the cubic spline kernels.

The impulse model did not perform as well as the edge model on the images tested. However the edge model gave almost optimal performance and was on par or better than the standard methods used for testing.

The significance of this work is that it provides a unifying framework that brings together both linear and cubic interpolators. Cubic spline interpolation is widely used because it is known to give good results. However it is based on an arbitrary function. Here we have shown that we can get an almost identical kernel, and similar results using a model that is derived explicitly from image characteristics and the imaging model.

The approach described in this paper can be readily extended to 2D images, however the analysis is a lot more laborious. A further extension of this work would be to include the lens point spread function and noise within the imaging model.

5 References.

[1] S.C. Park, M.K. Park and M.G. Kang, "Super-resolution image reconstruction: a technical overview", *IEEE Signal Processing Magazine*, 20(3), pp 21-36 (2003).

[2] L. Brown, "A survey of image registration techniques", *ACM Computer Surveys*, 24(4), pp 325-376 (1992).

[3] A. Gilman, D.G. Bailey, S.R. Marsland, "Noise characteristics of higher order predictive interpolation for sub-pixel registration", appearing in *IEEE International Symposium on Signal Processing and Information Technology* (2007).

[4] A. Gilman and D.G. Bailey, "Near optimal non-uniform interpolation for image super-resolution from multiple images", *Image and Vision Computing New Zealand*, Great Barrier Island, New Zealand, pp 31-36 (2006).

[5] MATLAB On-line Documentation, "interp1", http://www.mathworks.com/access/helpdesk/help/techdoc/ref/interp1.html, visited on 20/11/2007

[6] R. Franke, "Scattered data interpolation: tests of some methods", *Mathematics of Computation*, 38(157), pp 181-200 (1982).