Empirical Filter Estimation for Subpixel Interpolation and Matching

Bill Triggs

CNRS-INRIA, 655 avenue de l'Europe, 38330 Montbonnot, France.
Bill.Triggs@inrialpes.fr ◇ http://www.inrialpes.fr/movi/people/Triggs

Abstract

We study the low-level problem of predicting pixel intensities after subpixel image translations. This is a basic subroutine for image warping and super-resolution, and it has a critical influence on the accuracy of subpixel matching by image correlation. Rather than using traditional frequency-space filtering theory or *ad hoc* interpolators such as splines, we take an empirical approach, finding optimal subpixel interpolation filters by direct numerical optimization over a large set of training examples. The training set is generated by subsampling larger images at different translations, using subsamplers that mimic the spatial response functions of real pixels. We argue that this gives realistic results, and design filters of various different parametric forms under traditional and robust prediction error metrics. We systematically study the performance of the resulting filters, paying particular attention to the influence of the underlying image sampling regime and the effects of aliasing ("jaggies"). We summarize the results and give practical advice for obtaining subpixel accuracy.

Keywords: image filtering, subpixel interpolation, super-resolution, aliasing, subpixel matching.

1 Introduction

What is the best way to obtain subpixel accuracy from images? – Ultimately, it is a question of which feature extraction, filtering or interpolation scheme to use. Interpolation schemes are often motivated theoretically [18, 9, 10], either as finite-width approximations to the infinite 'sinc' filters that exactly interpolate band-limited signals in Nyquist sampling theory[1], or in terms of convenient parametric forms such as cubic splines. But at bottom the question is empirical. Real images are neither cubic nor strictly band-limited to the pixel spacing. Nyquist and spline theory neither give us optimal interpolators for them, nor tell us which of the many suboptimal approximations to use. These issues exist in 1D but become even thornier for images, where multidimensional sampling

artifacts — notably the 'aliasing' of non-grid-aligned edges into staircases of discrete steps ("jaggies") — can seriously degrade feature detection and geometric precision.

In default of an effective theory, we treat these issues empirically, designing accurate interpolators by minimizing their prediction errors over sets of training images containing subpixel translations. Our current implementation is oriented towards quantitative matching (subpixel correlation) rather than human perception: we choose intensity-related error metrics (*e.g.* L1, L2) rather than perceptual ones (*e.g.* [12]); we consider geometric as well as photometric accuracy; and we pay particular attention to the (strong) influence of the underlying pixel spatial response function. Our approach could also be used for super-resolution [16, 3, 5, 4], but here we use individual not multiple images, and we aim to predict what the original camera would return if shifted, not an enhanced image. An illuminating 1D analytic study complementary to our 2D empirical one is [15]. For a unified "information optimizing" approach to sampling and reconstruction, see [7]. For subpixel reconstruction based on learned "codebooks" see [6, 1].

Our main aim was to establish 'good working practice' for accurate subpixel image manipulation, side-stepping the bewildering range of methods available for filter design [9, 10]. Forms like splines are essentially heuristic, and we do not accept that strict band-limitation and sinc interpolation are the ideal approach for vision, to be approximated as well as the available computational power allows. The high frequency spectra of natural images are dominated by phase-coherent $1/f$ components produced by step edges. Bandwidth truncation of these produces significant high frequency 'ringing', which disturbs both the human eye and accurate computer vision algorithms. As figure 1 shows, abruptly band-limited images simply do not "look right": oscillations propagate out from each edge, making featureless surfaces look textured and generally confusing the signal[2]. If band-limited images are to be used, the downstream image processing modules must be capable of handling such artifacts.

Notation: PRF stands for Pixel (spatial) Response Function (§2). Image points are denoted $\boldsymbol{x} = (x, y)$ with input pixels (centres) at integer coordinates. w denotes filter

---

[1]We assume familiarity with basic sampling theory. Sampling a continuous signal on a discrete grid folds ('aliases') high spatial frequencies down to their fractional parts (in cycles per grid unit), and thus confuses the signal. Signals limited to the **Nyquist frequency** band $[-\frac{1}{2}, \frac{1}{2}]$ (no wavelengths less than 2 pixels) have no aliasing and hence can be reconstructed exactly. Bandwidth limitation / band-limited signal reconstruction can be implemented by continuous / discrete convolution against a **sinc function** $\mathrm{sinc}(\pi x) \equiv \sin(\pi x)/(\pi x)$, whose abruptly truncated Fourier transform (1 for $|f| < \frac{1}{2}$ and 0 elsewhere) but infinite, slowly decaying oscillating tails in x make it expensive to implement accurately by direct convolution.

[2]These artifacts are more visible on the computer than in print. Here they are due to abrupt bandwidth truncation alone, not finite filter width or image boundary effects (we used carefully windowed FFT on much larger images). Poorly truncated sinc filters give even worse ringing.
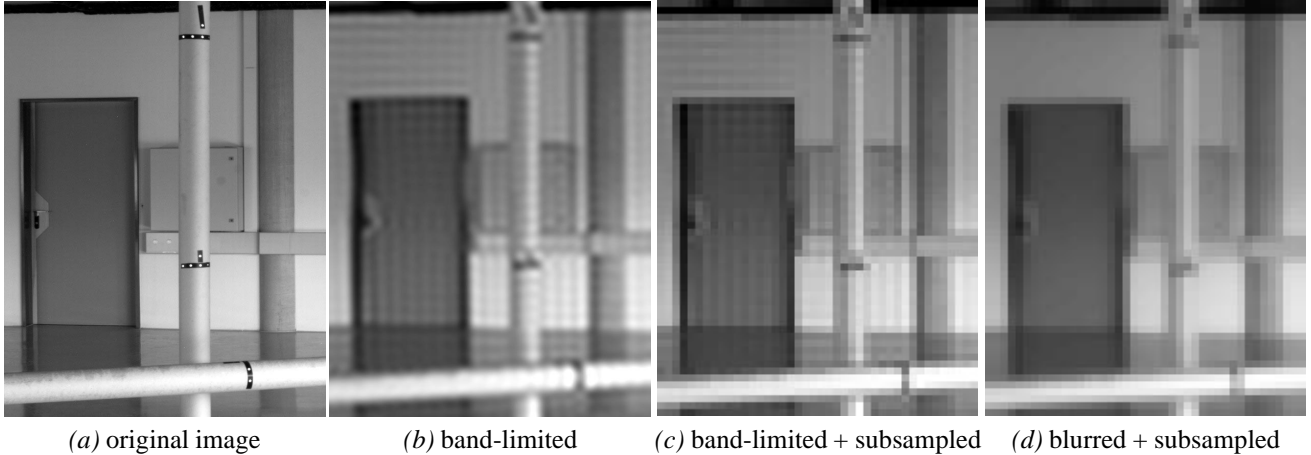
| *(a)* original image | *(b)* band-limited | *(c)* band-limited + subsampled | *(d)* blurred + subsampled |

Figure 1: The effects of abrupt bandwidth limitation. An original image *(a)*, limited to $1/15$ of the bandwidth *(b)* then decimated (point-subsampled) $15\times$ *(c)*. The band-limited images *(b,c)* have characteristic "ringing" artifacts: Nyquist frequency waves propagate outwards from each edge making smooth surfaces look striped or plaided, especially between parallel edges. Blurring *(a)* with a $\sigma = 7$ Gaussian before subsampling gives a slightly less sharp image *(d)*, but no ringing.

half-width: a half-width w interpolator reconstructing subpixel location (x, y) accesses all integer positions (i, j) with $|x - i|, |y - j| < w$. §2 discusses our method of generating training data, §3 comments on aliasing, §4 sketches our filter design method, §5 presents our experimental findings, and §6 summarizes.

2 Image Formation & Subsampling

To estimate subpixel interpolation filters, our training method (§4) needs a large body of training images with accurately labelled subpixel shifts. Suitable data could perhaps be collected experimentally, but this would be time consuming and error-prone. Instead, we have preferred to synthesize suitable data by carefully subsampling real source images. This allows us to rapidly generate a large body of training examples with guaranteed-accurate subpixel shifts, and it can synthesize measurements corresponding to any given camera response (*i.e.* PRF, see below), which gives us great flexibility in filter design. The danger is that subsampled images may be "unrealistic" — too unrepresentative of real scenes to produce useful interpolators. This section argues that appropriate subsampling does capture the relevant aspects of real scenes. The argument relies on two empirical properties of the underlying continuous input images:

(*i*) Within small enough regions, many natural scenes are nearly scale invariant, with smooth featureless power-law spectra. Zooming out does not change their local statistics: local windows on zoomed scenes look similar to local windows on unzoomed ones. In particular, rescaling step edges changes neither their appearance nor their $1/f$ spectra, so images dominated by abrupt transitions between uniform regions are locally scale-invariant, and interpolators fitted to such data should remain valid for other edge-dominated scenes.

Zooming does change the ratio of uniform regions to edge-containing ones, but this has little effect on the results as uniform regions are uninformative for interpolator training (they constrain only the average of the filter coefficients).

(*ii*) The trained interpolators depend mainly on characteristics of the input spectra within a few octaves of the pixel sampling frequency: higher frequencies are usually too strongly attenuated to cause much aliasing and too phase-sensitive to reconstruct[3], and lower ones look like constants within the filter's limited window. Hence, the un-subsampled discrete source images contain all of the frequencies needed to synthesize realistic images of a zoomed-out scene, including sampling and aliasing effects.

These properties suggest that subsampling should be a useful strategy for synthesizing training data provided that we arrange for the subsampled images to mimic as closely as possible those that would have been produced by the same camera looking at a rescaled (zoomed out) scene, including all signal degradation, sampling and aliasing effects. To approximate this we use a simplistic local model of image formation. We assume that pixels are identical; that each responds linearly to the total light flux falling on it which is in turn a linear function of the underlying scene luminance; and that this linear process is shift-invariant: if the scene luminance is represented in image plane coordinates as an 'ideal' image $L(\boldsymbol{x})$, a pixel at $\boldsymbol{x}_0$ responds with $R(\boldsymbol{x}_0) = (P * L)(\boldsymbol{x}_0) \equiv \int_{\boldsymbol{x}} P(\boldsymbol{x}_0 - \boldsymbol{x}) \, L(\boldsymbol{x}) \, d\boldsymbol{x}$, where "$*$" denotes convolution and the **Pixel Response Function (PRF)** $P(\boldsymbol{x}_0 - \boldsymbol{x})$ represents the combined effects of scattering, blurring, diffraction, other signal degradations, and flux integration across the pixel's recep-

---

[3]Linear interpolators are convolution-like but do not necessarily preserve sinusoids, so they can synthesize frequencies not present in their input. They can be viewed as convolvers acting on densified signals constructed by intercalating zeros between the input samples, which extends their spectra.
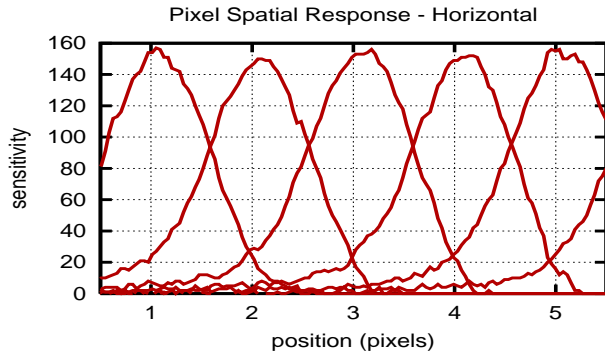
Figure 2: Experimental horizontal pixel responses (PRF's) for an analog video camera. These give the light sensitivity of the pixels as the signal moves across them.

tive zone. Clearly this model is only approximate — real pixel responses are nonlinear (saturation, clipping, quantization), and the various degradations depend on 3D and image position, wavelength, optics and geometry[4] — but it will suffice as a local model for predicting "average" subpixel behaviour. Interpolators that adapt to local imaging conditions are left for future work.

To predict the image R' of the rescaled scene, we apply the desired PRF P at the new scale rather than the old one: $R'(\boldsymbol{x}) = (P * L_\lambda)(\boldsymbol{x}) = (P_{1/\lambda} * L)(\lambda \boldsymbol{x})$ where $L_\lambda(\boldsymbol{x}) \equiv L(\lambda \boldsymbol{x})$ is the reduced scene ($\lambda > 1$) and $P_{1/\lambda}(\boldsymbol{x}) \equiv P(\boldsymbol{x}/\lambda)$ is the expanded PRF. In practice this amounts to simply convolving the discrete source image with a sampled version of the expanded PRF before subsampling: the original PRF is not eliminated, but for $\lambda \gtrsim 4$ it is so much narrower than the expanded one that its additional smoothing effect is negligible.

Typical PRF's (see fig.2) are around one pixel wide, with a form dominated by pixel geometry for narrower PRF's and optics (blurring, diffraction) for wider ones. §3 shows how to estimate the PRF of a real camera. §5 experiments with a number of idealized PRF's in order to determine which aspects of PRF form are important for subpixel accuracy.

3 Aliasing

Digitization converts smooth sloping edges into staircases of more or less abrupt steps or "jaggies", an effect (incorrectly) called "aliasing". The experiments below show that for well-focused images at typical daylight noise levels, aliasing is

---

[4]Apart from blur, pixel geometry is the main PRF determinant. "Interline" CCD's include shielded channels to prevent smearing caused by incoming light during readout shifting, so only about 40–70% of the pixel area is light sensitive, while "full frame" CCD's use the light collectors themselves for readout and have sensitive areas nearer 100% [17, 10]. Chip-surface microlenses and colour filters alter the PRF, and position dependent variations occur near the borders of large chips where the incoming rays strike the surface quite obliquely. Sharpening filters are often included in the electronics to reduce blur, but cause asymmetry and ringing that may lead to clipping near strong edges. Saturation and clipping severely degrade the geometric precision, so for precise applications it is best to underexpose the images and to switch off any deblurring filters.
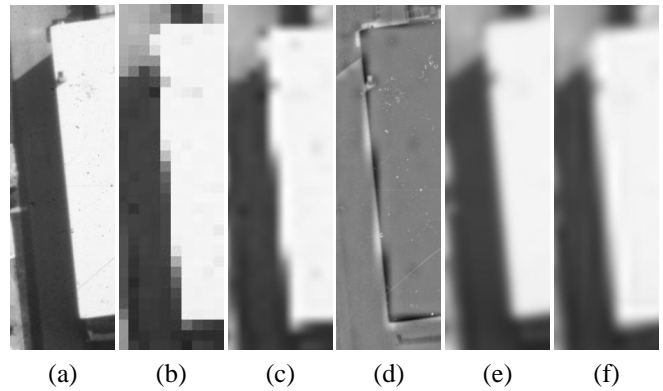

Figure 3: Aliasing along an edge due to poor sampling. When the original image (a) is decimated $15\times$ it shows clear jaggies (b). Even an optimal reinterpolation filter can do little to recover from these (c). The prediction error (d) is concentrated along strong edges, in phase with the jaggies. A more smoothly (0.5 pixel Gaussian) subsampled image (e) has no visible jaggies and can be reconstructed (f) with far less error.

easily the dominant source of error in subpixel interpolation. Fig.3 illustrates this on the extreme case of a point-subsampled image. Even well designed interpolation filters can do little to recover from aliasing once it is present. In fact, *no* finite-window filter can eliminate all aliasing. As an edge moves forwards one pixel, its jaggies move sideways by one cycle (one "jag") by a Moiré effect. As the edge becomes more closely aligned with one of the coordinate axes, its jaggies become ever more widely spaced and move ever more rapidly sideways. Eventually, they are more widely spaced than the filter window and the filter has no way of predicting when they will pass through it.

The Moiré properties of aliasing along edges provide an easy means of measuring the effective horizontal and vertical PRF's of the camera. If we take a shallow step edge and move parallel to the coordinate axis it is nearly aligned with, the step gradually encroaches on the pixels we cross as we move. Hence, these pixel intensities effectively give a fine sampling (with spatial resolution $1/$(aliasing period of edge)) of the cumulative response of a pixel as an edge moves across it. Taking derivatives along the line of samples gives the non-cumulative pixel response function, *i.e.* the response as a line edge moves across the pixel. For example, in fig.3 (b), sampling along a vertical line gives a fairly abrupt step as the line hits its jaggie, and differentiating this gives the vertical PRF, which is nearly a delta function. The results in fig.2 were obtained in this way.

4 Filter Design

Our design method for subpixel interpolators is extremely simple: we collect a *training set* of pairs of images with known subpixel relative displacements, choose a *parametric form* for the filter and an *error metric* for its prediction error,
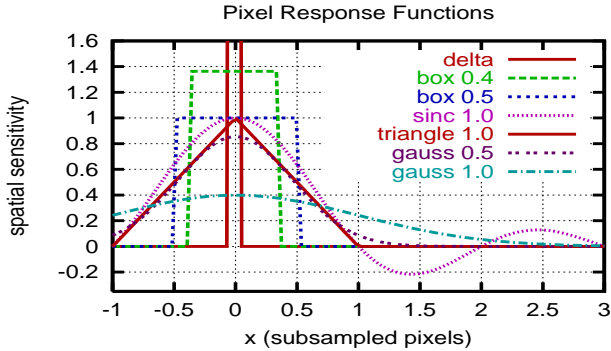
Figure 4: The pixel response functions for which filters were designed in this study.

and *numerically minimize* the filter's total prediction error on the test images over its parameters. Consider each element in turn.

Training images: Training requires a large set of accurately labelled subpixel data. Any images could be used. The study below subsampled large photogrammetric images to generate the training data. As discussed in §2, we believe subsampling to be adequately realistic, and it allows arbitrarily large and varied data sets to be generated quickly and reliably. It also allowed a detailed comparison of the effects of different PRF's, which would not otherwise have been possible. A $15\times$ subsampling factor provided reasonably dense interpolation and ensured that the PRF of the original source camera had negligible influence on the results. We considered only grayscale images as the non-collocated sampling of Bayer-style $^{\text{RG}}_{\text{GB}}$ colour arrays greatly complicates subpixel issues.

Parametric filter: We designed filters from several different parametric families. All were based on square windows centred on the inter-pixel square containing the subpixel position being predicted. The mask of a half-width w filter includes exactly w input pixels on each side. Half-width 1 filters access just the four corners of the 1 pixel square containing the point being predicted, and in practice tend to approximate bilinear interpolation. Our simplest parametrization leaves the weights at all positions and shifts as free parameters. Alternatively, we can fit a separable filter, where each weight is the product of the corresponding x-filter and y-filter ones. Finally, we can parametrize each inter-pixel section of the separable filters as a separate cubic, which further reduces the number of parameters, couples the weights at different shifts, and allows subpixel shifts not in the original training set to be generated. We did not implement continuity constraints between the cubic sections because the optimal filters often have small discontinuities in practice. We can also require the impulse response to be symmetric if desired, but real PRF's are often asymmetric, especially along scan lines.

Error metric: Our designs minimize intensity (grey-level) prediction error rather than, *e.g.* perceptual or feature location

metrics. The errors are summed over all resampled pixels and all subpixel shifts being considered. Let δI_i be the difference between the observed and the predicted intensity at pixel i and σ be a robustness-scale parameter. We designed filters under the following error norms. L2: classical least squares, $|\delta I_i|^2$. L1: least absolute value, $|\delta I_i|$. Lorentz: $\log\left(1 + \frac{1}{2}|\delta I_i/\sigma|^2\right)$. Smoothed L1: $\sqrt{1 + \frac{1}{2}|\delta I_i/\sigma|^2} - 1$.

Optimization method: The most conspicuous feature of the optimization problem is the huge number of measurements available — more than 10^7 for the designs given below. It often has many parameters too. Separable filters have comparatively few, but the largest non-separable filters tested below — half-width 6 with $15\times$ subsampling — have $(2 \cdot 6 \cdot 15)^2 = 32400$ free parameters in 255 groups of 144 (the groups being independent if no inter-shift smoothness constraints are enforced — and none seem to be needed). However, the data is very strong, so apart from its size the optimization problem is relatively benign. Given these characteristics, we chose to use Liu & Nocedal's elegant LBFGS (limited memory quasi-Newton) optimizer [11, 13] from Netlib (http://www.netlib.org). The required values and gradients of the error function with respect to the filter parameters are calculated by a predict-and-accumulate cycle through the test images to evaluate gradients with respect to the parameters of the non-separable filter, followed by chain-rule reduction to a smaller parameter space if required. We use LBFGS for linear (non-separable filters in the L2 norm) and non-smooth (L1 error norm) problems, as well as for the smooth nonlinear ones (separable filters or robust error norms) for which it was designed. On L1 problems, LBFGS works much better than a purpose designed simplex-method code we tested, which thrashed owing to the huge number of equations involved.

5   Experimental Results

We made a systematic study of subpixel interpolation filter design using the above methods. There is only space for a brief summary of what we learned. The results shown are based on the *Ascona Workshop* test images — eight 1800×1800 aerial views of a suburban scene, taken with a photogrammetric camera and a high-resolution colour film scanner (http://www.photogrammetry.ethz.ch/news/events/ascona/dataset/dataset.html). The images were converted to grayscale and subsampled $15\times$ with the desired PRF's at integer shifts between 0 and 15 (*i.e.* subpixel x and y shifts in $(0, 1...15)/15$), so the training PRF's are accurate to around $15\times$ the target resolution. The Ascona images have $\sigma \sim 2$ blur and some spots and scratches, but training on sharper, cleaner indoor images (six 3072×2048 images from IMETRIC using a Kodak DCS460 digital camera with $\sigma \sim 0.7$, *c.f.* fig.1) gave essentially identical results, as expected. (Full images not shown due to lack of space).

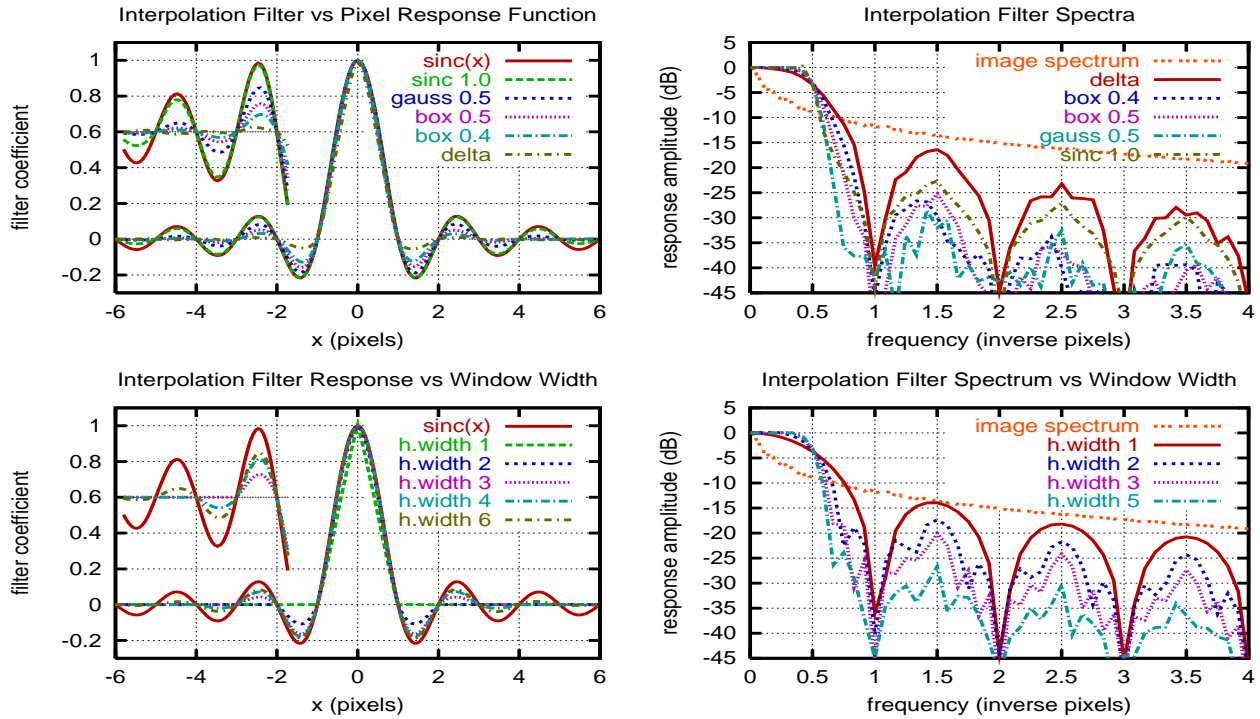**Design results:** We designed separable and non-separable

Figure 5: Optimal interpolation filter impulse responses (left) and spectra (right) for separable L2-norm filters designed for various pixel response functions (top, for half-width $w = 6$) and filter widths (bottom, for the gauss 0.5 PRF). A sinc function (left) and the image spectrum (right) are plotted for comparison.

$15\times$ interpolation filters of half-widths $w = 1...6$ pixels under the above four error norms for the following seven PRF's (see fig.4). *Delta:* unsmoothed decimation / point subsampling (by far the worst case for aliasing). *Box 0.4, 0.5:* box filters of half-width $5/15$ and $7/15$ representing ideal square pixels with sensitive areas of full widths $(2 \cdot 5+1)/15 = 0.73$ and 1.0 pixels, the former being representative of 'interline' CCD's, the latter of 'full frame' ones [17]. *Triangle 1.0:* width 1 triangular response representing ideal bilinear interpolation. *Gauss 0.5, 1.0:* Gaussian responses of $\sigma = 0.5, 1.0$ pixels, representing typical, and fairly large, amounts of optical blurring. *Sinc 1.0:* the ideal Nyquist-band-limiting PRF (may not be physically realizable owing to negative coefficients). Sinc subsampling used carefully windowed FFT to minimize truncation and boundary effects. Where not otherwise noted the following default parameters were used: training and test PRF, Gaussian 0.5; filter, separable, all entries free, half-width $w = 3$; error norm, L2. We train and test on the same (subsampled Ascona) images: validation against the IMETRIC set showed that the effects of overfitting were negligible.

Fig.5 shows how the impulse responses and spectra of the interpolators vary with design PRF and width. All of the designs have sinc-like oscillations, but except for the sinc PRF these decay much more rapidly than a sinc or classically-windowed sinc function. The Delta PRF interpolator is nearly triangular with very small side-lobes, while those for the broader PRF's have 2–3 visible side-lobes. Reducing the design width progressively suppresses the side-lobes until the filters become triangular at $w = 1$. Half-widths 2–4 are probably the best compromise in practice. The spectra show the same story, with all filters having side-lobes significantly smaller than a truncated sinc of the same width[5]. As expected, the spectra of the training images are relatively featureless at these scales, so the designs are insensitive to the particular images used.

It is instructive to view the sinc PRF interpolators as windowed versions of the ideal sinc interpolator. Dividing by $\text{sinc}(x)$ gives the implied empirically optimal windowing functions (fig.6). The recovered window functions are broader and more abrupt than most classical ones (the relatively broad Welch $(1 - (x/w)^2)$ and Lanczos $(\text{sinc}(x/w))$ ones are shown for comparison). As a result, the sinc PRF interpolators are far from band-limited: their spectra are actually *broader* than those resulting from most other PRF's. The optimal windows also descend by abrupt steps, rigidly scaling each side-lobe. This gives the optimal interpolators small but significant derivative discontinuities at each zero crossing, especially where the last two lobes drop to zero in two $\sim 40\%$ steps. In effect, *the input data is windowed, not the interpo-*

---

[5]The slight bumpiness of the spectra is due to FFT aliasing — the Nyquist limit for $15\times$ interpolators is only 7.5 (pixels)$^{-1}$ — and noise — $\mathcal{O}(10^7)$ training pixels leaves $\mathcal{O}(\sqrt{10^{-7}}) \sim -35$ dB noise.
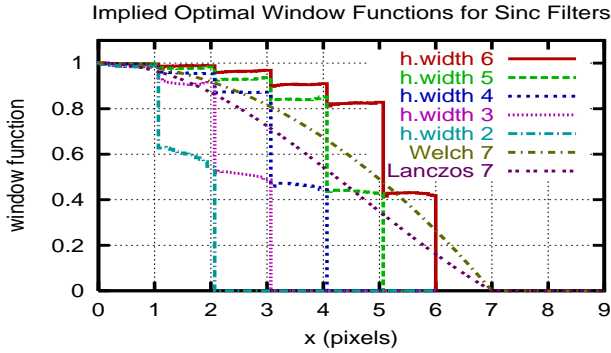
Figure 6: The ratios $\mathrm{filter}(\boldsymbol{x})/\mathrm{sinc}(\boldsymbol{x})$ for $w=2...6$ sinc PRF filters give the implied optimal windowing functions for sinc interpolators of these widths.
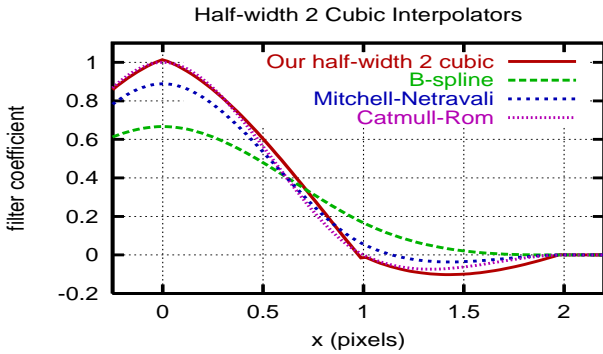

Figure 7: The optimal cubic filters (here $w=2$ for the Gauss 0.5 PRF) have small derivative discontinuities and larger side lobes than either the cubic-convergent Catmull-Rom spline or Mitchell & Netravali's visually preferred "$(\frac{1}{3},\frac{1}{3})$" design [12].

lator: the interpolators for all fractional shifts share the same input points and the same windowing.

The fitted bicubic interpolators tell the same story for the other PRFs. They have significantly larger side-lobes than commonly recommended bicubics such as Catmull-Rom or Mitchell-Netravali splines [12], and again small but distinct derivative discontinuities (fig.7). Optimal interpolation does *not* imply either gentle windowing or high-order smoothness.

Filters designed under different error norms are for the most part nearly identical, although L1 designs typically have slightly smaller side lobes than L2 ones, with the two robust norms intermediate between these two. The differences are largest for the Delta PRF. Aliasing ensures that the input data contains significant numbers of outliers, but these are distributed fairly symmetrically so they do not upset the L2 fits too much. The 2D designs have small but consistent non-separabilities (about 5% of amplitude for most PRF's, mostly in the central ± 1 pixel square), but this does not seem to give them a noticeable performance edge over their separable counterparts, except perhaps for the Delta PRF.

Test results: Now consider the experimental performance of

the designed filters. Our tests use large images subsampled as for the training images. We consider two aspects of sub-pixel performance: intensity interpolation error (which the filters were designed to minimize) and feature location error. Intensity testing is self evident: we interpolate the test images and compare the result to the original shifted versions, accumulating errors in some chosen error metric (here L1). The location tests were designed to study the "ground truth" performance of subpixel correlation matching (not, *e.g.*, feature detection). We defined a set of strong isolated test features with accurately-known subpixel locations by selecting 100–150 "points of interest" — locations at which correlation matching of the image against itself gives high spatial accuracy in all directions — in each unsubsampled test image. For this we used a Harris (Lucas-Kanade-Harris-Förstner) detector with inflated scale parameters (so that the detected features should be strong points of interest in the subsampled images), and suppression parameters set to eliminate all spurious responses along edges and in textured regions. This provided a test set that was independent of the PRF and shift used (real 3D features exist independently of these!), and that had minimal aliasing artifacts (in particular, no spurious responses along edges). Visual inspection confirmed the high quality of the chosen points. The location tests used the subsampled locations of these points as correlation centres. For each centre and shift, we interpolated the image and made an exhaustive search of the correlation at all $1/15^{th}$-pixel locations within a window of $\pm 20/15$ pixels around the known output centre. The best result found was taken as the match, and the resulting translational errors were recorded and analyzed. To minimize edge effects, correlations were calculated by integrating the L2 (squared) intensity prediction error over a $\sigma=2.0$ Gaussian window. Optionally, independent noise was added to the base and target images before matching. This approach is idealistic in using near-perfect features with no geometric or photometric distortion between the base and target windows, but it allows us to study just one thing at a time, here aliasing and subpixel accuracy.

Fig.8 summarizes the performance of the designed filters. The salient point is the overwhelming influence of the test image PRF on all aspects of subpixel performance. Aliasing along edges is by far the dominant source of error in these plots, and the main factor controlling aliasing is the test PRF: the narrower it is, the more aliasing and error there is. The Delta PRF (decimation) is particularly bad in this respect and dominates the error plots. In contrast, as the top row shows, the design PRF for the interpolation filter has relatively little influence. Basically, filters designed for any PRF of similar width give similar results. The plots in the bottom row show that there is little advantage in extending the filter half-width beyond about 3 pixels. Spatially, the errors are largest at half-pixel shifts, as would be expected.

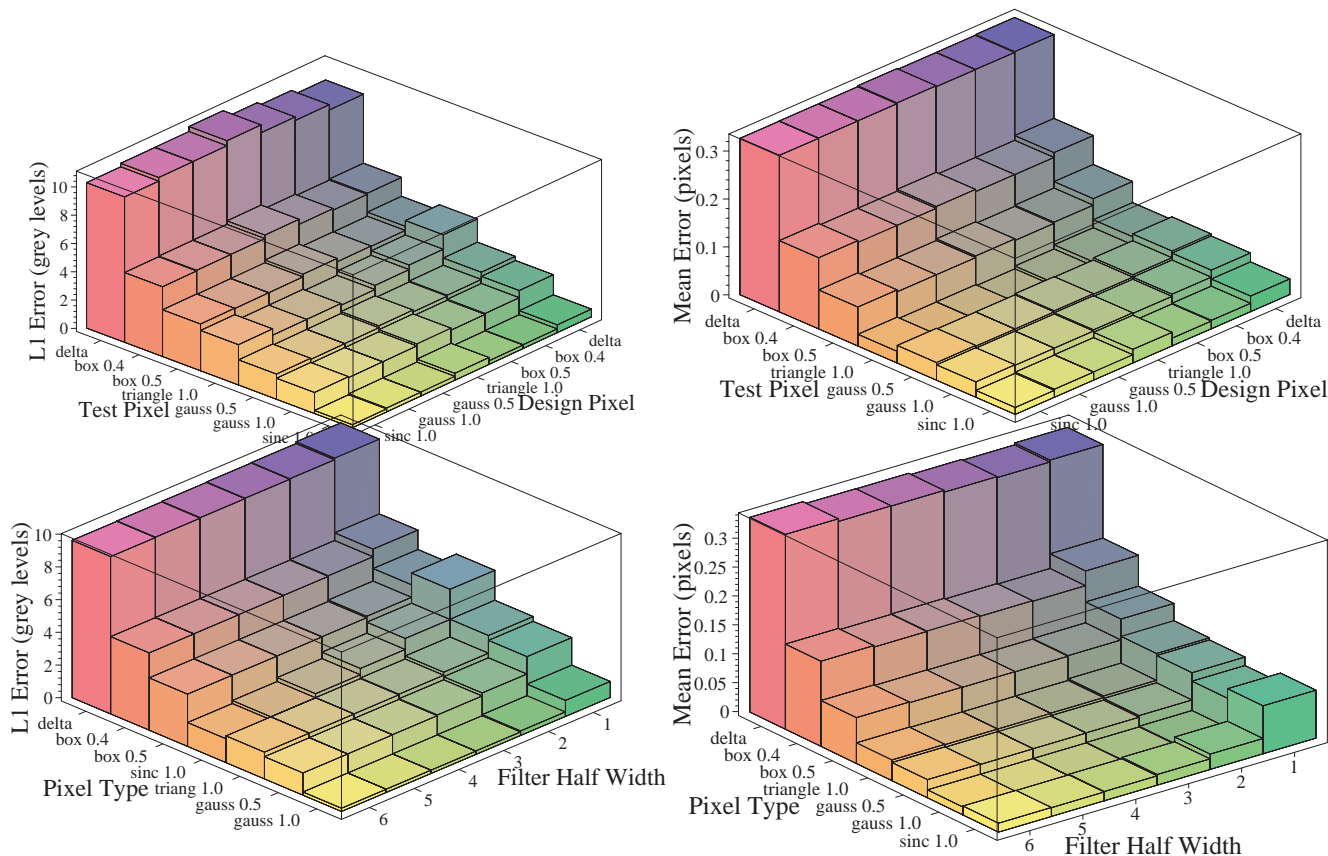Given the extent to which smoother PRF's reduce aliasing,

Figure 8: Interpolation error (left) and location error (right) versus (top) design and test pixel type and (bottom) filter half-width. Unless otherwise noted all filters are tested on the PRF's they were designed for.

we can ask whether it is also useful to smooth the images *after* sampling. Unsurprisingly, post-sampling smoothing of both source and test images does reduce interpolation error. Fig.9 suggests that small amounts ($\sigma \approx 0.5$ pixels) of post-smoothing also slightly reduce the location error in correlation matching, both for noiseless (no added noise) and quite noisy images. However the benefit is minor and is probably due to the effective increase in the size of the correlation integration window. Post-smoothing greater than about 0.8 pixels rapidly degrades spatial accuracy. Note also the poor performance of the sinc PRF filter at high noise levels — the extended oscillations of sinc filters give them larger squared integrals, and hence more noise accumulation, than comparable Gaussians.

6   Summary and Conclusions

For everyday cameras and noise levels, aliasing ("jaggies") is the dominant cause of accuracy loss in subpixel interpolation and matching, not random noise. Aliasing is determined mainly by the Pixel spatial Response Function (PRF). Over-narrow PRF's have large aliasing and much lower accuracy. Image processing after sampling does little to reduce the effects of aliasing, but a small amount of additional optical blurring *before* sampling can sometimes improve accuracy considerably. The effective PRF should ideally have a standard deviation of around 0.5 pixels but its exact form is not critical. Visually, this corresponds to just enough blurring to make jaggies along strong edges disappear, *c.f.* fig.3(e).

Whatever the PRF, it is easily measured by taking derivatives along strong nearly-axis-aligned edges, and a custom subpixel interpolation filter can then be designed for it. Optimal interpolators of any desired parametric form can be designed by explicitly minimizing their total L2 or robust interpolation error over a set of training images. Suitable training images can be synthesized by subsampling larger ones, using the desired design PRF at different relative shifts. The resulting optimal filters have rapidly decaying sinc-function-like oscillations, separable filters of half-width 2–3 being fully sufficient for most practical applications. The filter forms depend on the widths of their design PRF's, but are relatively insensitive to the exact PRF shape, the error norm and the training images used. Filters designed for different but broadly similar PRFs give similar interpolation and matching performance, so the design and application PRF's do not need to be closely matched. Smooth windowing and high order continuity are *not* needed for optimality: the optimal subpixel interpolators have small derivative discontinuities at their zero crossings, and their implied windowing functions have abrupt steps there.

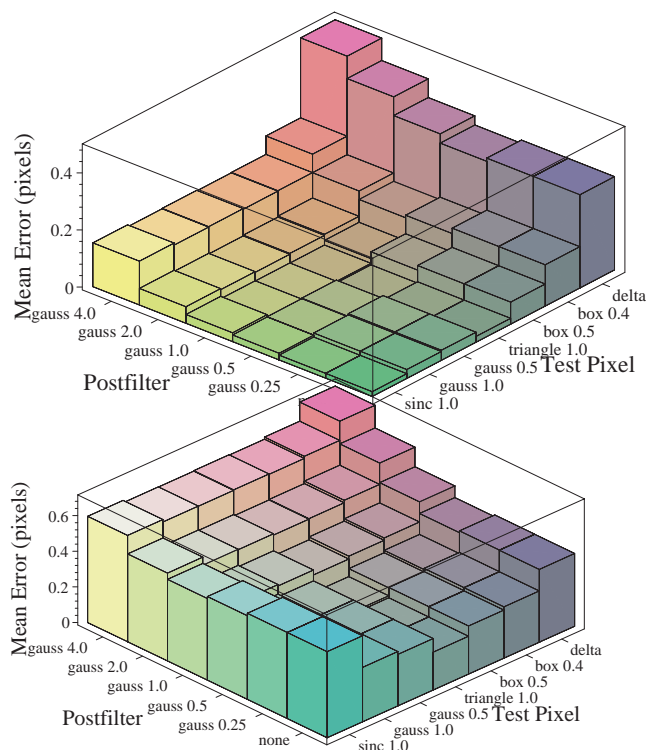**Future work:** A useful minor extension would be to design

Figure 9: Location error versus test pixel type and amount of post-sampling smoothing, for the gauss 0.5 filter. The top plot is with no added noise, the bottom one with Gaussian noise of $\sigma = 20$ gray-levels.

interpolators for half-windows, to allow optimal signal reconstruction near occlusion boundaries. Other low-level image operators such as feature detectors would probably also benefit from explicit empirical performance optimization, the main barrier being (as here) the lack of suitable training data.

We only considered linear convolution filters here but the approach extends easily to nonlinear ones, *e.g.* linear combinations of nonlinear functions of pixel intensities or intensity differences... It would be interesting to see whether such filters could provide better resistance to aliasing. Kernel-based learning methods such as Support Vector Machines could perhaps also be used [14], although speed might be a problem and our initial attempts to train an SVM filter failed owing to the huge number of measurements involved. Despite the many advantages of linear sinusoidal sampling theory, we do feel that some sort of nonlinear theory built from steps, corners and their integrals (ramps, *etc.*) is needed for computer vision, as the underlying signal is to a large extent built out of such elements at sparse but arbitrary subpixel positions and orientations, rather than sinusoids. At present we have no such theory, but psychophysical data and recent results on independent components analysis and kernel based learning all seem to point in this direction [2, 8, 14].

References

[1] S. Baker and T. Kanade. Limits on super-resolution and how to break them. In *Int. Conf. Computer Vision & Pattern Recognition*, volume 2, pages 372–379, 2000.

[2] A.J. Bell and T.J. Sejnowski. The 'independent components' of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997.

[3] T. Boult and G. Wolberg. Local image reconstruction and sub-pixel restoration algorithms. *Computer Vision, Graphics & Image Processing*, 55(1):63–77, 1993.

[4] D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *Int. Conf. Computer Vision & Pattern Recognition*, pages 885–891, June 1998.

[5] M.C. Chiang and T.E. Boult. Efficient super-resolution via image warping. *Image & Vision Computing*, 18(10):761–771, 2000.

[6] W. Freeman, E. Pasztor, and O. Carmichael. Learning low-level vision. *Int. J. Computer Vision*, 40(1):25–48, 2000.

[7] F.O. Huck, C.L. Fales, and Z. Rahman. An information theory of visual communication. *Phil. Trans. Roy. Soc.*, A 354:2193–2248, 1996.

[8] A. Hyvärinen and P.O. Hoyer. Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720, 2000.

[9] B. Jähne, editor. *Handbook of Digital Image Processing for Scientific Applications*. CRC Press, Florida, 1997.

[10] B. Jähne, H. Haußecker, and P. Geißler, editors. *Handbook of Computer Vision and Applications*. Academic Press, 1999.

[11] D. Liu and J. Nocedal. On the Limited Memory BFGS method for large scale optimization. *Mathematical Programming*, B 45:503–528, 1989.

[12] D. P. Mitchell and A. N. Netravali. Reconstruction filters in computer graphics. *Computer Graphics (SIGGRAPH'88)*, 22(4):221–228, 1988.

[13] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer-Verlag, 1999.

[14] C. Papageorgiou, F. Girosi, and T. Poggio. Sparse correlation kernel analysis and reconstruction. Technical report, MIT A.I. Lab, 1998. Memo 1635 (CBCL Memo 162).

[15] S. K. Park and R. A. Schowengerdt. Image sampling, reconstruction and the effect of sample-scene phasing. *Applied Optics*, 21(7):3142–3151, 1982.

[16] S. Peleg, D. Keren, and L. Schweitzer. Improving image resolution using subpixel motion. *Pattern Recognition Letters*, 5:223–226, 1987.

[17] M.R. Shortis and H.A. Beyer. Sensor technology for digital photogrammetry and machine vision. In K.B. Atkinson, editor, *Close Range Photogrammetry and Machine Vision*, pages 106–155. Whittles Publishing, Roseleigh House, Latheronwheel, Caithness, Scotland, 1996.

[18] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1990.