GRID FILTERS FOR LOCAL NONLINEAR IMAGE RESTORATION

Todd L. Veldhuizen and M. Ed Jernigan

Dept. of Systems Design Engineering, University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
tveldhui@seurat.uwaterloo.ca

ABSTRACT

We describe a new approach to local nonlinear image restoration, based on approximating functions using a regular grid of points in a many-dimensional space. Symmetry reductions and compression of the sparse grid make it feasible to work with eight-dimensional grids as large as 14^8. Unlike polynomials and neural networks whose filtering complexity per pixel is linear in the number of filter coefficients, grid filters have $O(1)$ complexity per pixel. Grid filters require only a single presentation of the training samples, are numerically stable, leave unusual image features unchanged, and are a superset of order statistic filters. Results are presented for blurring and additive noise.

1. INTRODUCTION

Global iterative filters for image restoration can produce excellent results, but tend to be computationally expensive. If speed is a concern, local nonlinear filters may provide a faster alternative. Local nonlinear filters restore an image one pixel at a time, estimating the original pixel value s_0 using a nonlinear function $F(x_0, \ldots, x_{n-1})$ of pixels from the degraded image. For example, a 3x3 filter would have as inputs the pixel to be restored (x_0) and its eight neighbours:

x_8	x_4	x_5
x_3	x_0	x_1
x_7	x_2	x_6

Median filters, order statistic filters [2], and Lee's local statistics filter [5] are well-known examples of such filters. For a signal and degradation process satisfying certain weak conditions, a function F which is optimal in the minimum mean-squared error (MMSE) sense exists. This MMSE-optimal filter tends to have no computationally reasonable form, so it must be approximated, for example with feedforward neural networks or Volterra series (polynomials). These approaches suffer from a serious drawback: training and filtering time per pixel is linear in the number of filter coefficients (network weights or Volterra coefficients), with the result that good nonlinear restoration filters tend to be slow. Polynomial filters are prone to additional problems with numerical stability.

In this paper we explore a new approach which combines a point estimator for flat regions with a nonlinear detail filter. The nonlinear detail filter approximates the MMSE-optimal F using a regular grid of points. For example, for a 3x3 filter the function F can be approximated by a 9-dimensional grid of points. Each grid point has an associated coefficient which gives the value of F at that point. Interpolation is used to handle inputs which don't coincide with a grid point.

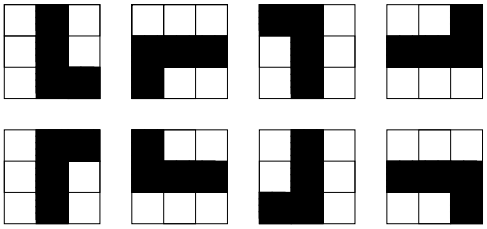


Figure 1: Eight orientations of a 3x3 window which should be treated the same by a filter with orientation-invariance.

2. THEORY AND IMPLEMENTATION

The filter has this basic structure:

$$\hat{s_0} = F(\mathbf{x}) = \beta(\mathbf{x}) F_1(\mathbf{x}) + (1 - \beta(\mathbf{x}))\, F_2(\mathbf{x}) \qquad (1)$$

where $\hat{s_0}$ is the estimate of the original pixel value, F_1 is a grid filter to handle detail regions, and F_2 is a point estimator used to restore flat regions. The vector $\mathbf{x}$ contains pixel values from the local neighbourhood. For additive white Gaussian noise, F_2 is a local average; for other noise distributions, order statistic filters provide better noise reduction [2].

The function $\beta(\mathbf{x})$ ranges between 0 for flat regions and 1 for highly detailed regions. The equation for β is borrowed from the Lee filter structure [5]. Given a variance estimate σ_x^2 for a local window, β is given by:

$$\beta = \max\left(\frac{\sigma_x^2 - \sigma_n^2}{\sigma_x^2}, 0\right) \qquad (2)$$

where σ_n^2 is an estimate of the noise variance.

2.1. Reducing the number of grid points

A grid filter with inputs from a 3x3 window requires a 9-dimensional grid. As the grid resolution is increased, the number of grid points gets large very quickly. Symmetry assumptions and a sparse representation are used to keep the number of grid points reasonable.

2.1.1. Orientation invariance

In many applications, the behaviour of the filter should be the same for different orientations of the same 3x3 window. Figure 1 illustrates 8 orientations of a 3x3 window which should be treated as equivalent for the purpose of restoring the central pixel. These 8 orientations define a permutation group on the grid coordinates;

assuming the function values to be invariant under this permutation group reduces the number of grid coefficients by a factor of (approximately) eight.

2.1.2. Signal mean invariance

In certain scenarios, the filter behaviour can be assumed independent of the local signal mean. If an amount δ is added to the input pixels, then the output should also shift by δ:

$$F_1(x_0+\delta,\ldots,x_{n-1}+\delta) = F_1(x_0,\ldots,x_{n-1}) + \delta \quad (3)$$

To exploit this property, set $\delta = -x_0$. Then, by (3),

$$F_1(x_0,\ldots,x_{n-1}) = x_0 + F_1(0, x_1 - x_0,\ldots,x_{n-1} - x_0) \quad (4)$$

This property eliminates one of the arguments of F_1, reducing the dimensionality of the grid by one. Signal mean invariance also turns the function F_1 into an *adjustment* applied to the degraded pixel value x_0 (4). This is crucial for numerical stability considerations, described later.

2.1.3. Reverse video invariance

Another symmetry is *reverse-video invariance*. Suppose I is an image, and denote by $\overline{I}$ the reverse video image. A filter $\mathcal{F}$ with reverse-video invariance satisfies $\mathcal{F}\overline{I} = \overline{\mathcal{F}I}$: filtering the reversed image gives the same result as reversing the filtered image. Let the maximum pixel intensity value be M. Then the reverse-video input pixels are $M - x_0$, $M - x_1$, ..., $M - x_{n-1}$. To have reverse-video invariance, F_1 must satisfy:

$$F_1(x_0,\ldots,x_{n-1}) = M - F_1(M - x_0,\ldots,M - x_{n-1}) \quad (5)$$

Applying the mean-invariance property (4) eliminates M from the equation:

$$\begin{aligned} F_1(x_0,\ldots,x_{n-1}) &= x_0 - F_1(0, x_0 - x_1,\ldots,x_0 - x_{n-1}) \\ &= x_0 + F_1(0, x_1 - x_0,\ldots,x_{n-1} - x_0) \end{aligned}$$

To exploit this invariance, the grid must be centered about (0,0,...,0). Then the grid possesses an anti-symmetry: for each grid point with value f_i, there is an antisymmetric grid point with value $-f_i$. This reduces the number of grid coefficients by a factor of two.

2.1.4. Sparse representation of the grid

The filter inputs $(x_0,\ldots,x_{n-1})$ form clusters corresponding to edges, flat regions, lines and ramps. Between these clusters are large volumes of empty space containing unused grid points. The number of grid coefficients can be drastically reduced by using a sparse representation which compresses these empty regions. Each grid point is stored in a hash table, keyed by integer grid coordinates. During training, grid points are added as required.

Figure 2 shows the number of grid points required for a certain 3x3 filter using various grid sizes. Without any reductions in grid size, an 8x8x...x8 grid would require 8^9 coefficients (about 130 million). Training such a filter would tax the abilities of a supercomputer with 32 Gb of memory. After applying symmetry reductions, the number of coefficients drops to roughly 1 million, which would require a high-end computer with about 1 Gb of memory to train. Using a sparse representation for the grid reduces the number of coefficients to a mere 23000. Such a filter can be trained in ten minutes on a typical workstation with 64 Mb of memory.

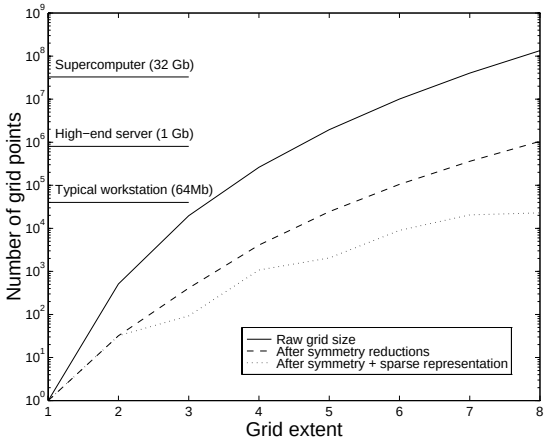


Figure 2: Number of grid points for a 3x3 filter

2.2. Interpolation method

Evaluating the grid filter $F_1(\mathbf{x})$ when $\mathbf{x}$ does not coincide with a grid point requires interpolation. For speed, this interpolation must involve as few grid points as possible. Piecewise linear interpolation achieves this: interpolating on a N-dimensional grid involves only $N+1$ grid points. For any point $\mathbf{x}$, one can find a bounding hypercube whose corners coincide with grid points. Let $\mathbf{x}'$ be the point after scaling into the unit hypercube $[0,1]^N$. The hypercube is sliced into $N!$ simplexes of the form

$$x'_{i_1} \le x'_{i_2} \le \ldots \le x'_{i_N} \quad (6)$$

where $(i_1, i_2, \ldots, i_N)$ is a permutation of $(1,\ldots,N)$. Over each simplex, the interpolation is linear and recovers the values of the grid points at the corners. The interpolation procedure produces a sparse vector $\mathbf{w}(\mathbf{x})$ of interpolation coefficients, one for every point in the grid. Only those coefficients corresponding to the N+1 contributing grid points are nonzero.

This scheme has an interesting relationship to order statistic filters [2]. Order statistic filters are also piecewise linear over regions of the form (6). If the grid is appropriately scaled and centered about the origin, it becomes a superset of order statistic filters. Performance is then guaranteed to be equivalent to, or better than, the optimal order statistic filter for the problem.

2.3. Training

Filter coefficients are chosen to minimize the expected mean-squared error. Let the vector of grid coefficients be $\mathbf{f}$, and $\mathbf{w}(\mathbf{x})$ the interpolation vector. The filter has the form:

$$F(\mathbf{x}) = \beta(x_0 + \mathbf{f}^T\mathbf{w}) + (1-\beta)F_2(\mathbf{x}) \quad (7)$$

The MMSE criterion function is:

$$J(\mathbf{f}) = E\left[(s_0 - F(\mathbf{x}))^2\right] \quad (8)$$

The expectation operator is approximated by summing over N training samples $\{s_0^j, \mathbf{x}^j\}$ drawn from pairs of pristine and degraded images. Minimizing J results in a linear system of equations $\mathbf{Af} = \mathbf{b}$, with:

$$\mathbf{A} = \sum_{j=1}^{N} \beta_j^2 \mathbf{w}_j \mathbf{w}_j^T \quad (9)$$

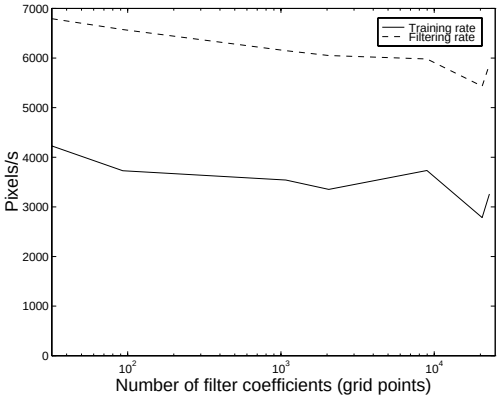


Figure 3: Training and filtering rates as a function of filter size

$$\mathbf{b} \quad = \quad \sum_{j=1}^{N} \beta_j \mathbf{w}_j \left(s_0^j - \beta_j x_0^j - (1-\beta_j) F_2(\mathbf{x}^j) \right)$$

where $\beta_j = \beta(\mathbf{x}^j)$ and $\mathbf{w}_j = \mathbf{w}(\mathbf{x}^j)$. The β^2 term in $\mathbf{A}$ and the β term in $\mathbf{b}$ weight the least-squares solution toward performance in detail regions. Since flat regions are handled by F_2, the grid filter can "concentrate its attention" on performing well in detail regions. The presence of the $(1-\beta)F_2$ term in $\mathbf{b}$ allows the grid filter to correct errors made by the local average estimate. The matrix $\mathbf{A}$ is sparse and symmetric, with roughly 20-30 nonzero elements per row. To avoid numerical instability arising from the approximation of the expectation operator, zero-order regularization [3] is used:

$$(\mathbf{A} + \lambda I)\mathbf{f} = \mathbf{b} \tag{10}$$

with λ chosen just large enough to ensure stability. When many choices of coefficients satisfy the optimality condition, the regularization selects the vector $\mathbf{f}$ with least norm. Since the signal-mean invariance assumption turns F_1 into an *adjustment* applied to the degraded pixel value, the regularization picks the filter which causes the least change in the degraded image. This ensures that unusual image features are passed unchanged by the filter. The regularization also drives the value of unused grid coefficients to zero, permitting use of the sparse representation. The sparse system of equations is solved using the Conjugate Gradient iterative method [1]. Note that unlike neural networks, grid filters require only a single presentation of the training samples.

3. RESULTS

3.1. Text images with additive noise

A filter using a 3x3 grid filter for F_1 and a 5x5 averager for F_2 was trained on images of three fonts (Helvetica, Times-Roman, and Courier) at 18 pt and 36 pt resolution, with intensity scaled to $[0, 255]$. The degradation process was Additive White Gaussian Noise (AWGN) with $\sigma^2 = 400$. Various grid sizes were used, from 2^8 up to 8^8. Figure 3 shows how training and filtering rates varied as the grid size increased (rates are for a 100 MHz RS/6000 workstation). In theory, these rates are asymptotically constant; the gradual decrease is due to cache effects. This is in sharp contrast to the filtering rates of feedforward neural networks and polynomials, which are inversely proportional to the number of filter coefficients. Figure 4 shows filter performance on a sample of 24 pt Palatino text (not in the training set). All grids larger than 3^8 outperformed the global Wiener filter and Lee's local statistics filter. Figure 6 shows a small portion of the restoration produced by the 8^8 grid filter. Figure 5 shows the average signal-to-noise ratio as a function of normalized spatial frequency (λ^{-1}) for the 8^8 grid filter result. In medium frequency bands ($\lambda^{-1} \approx 0.1$), the 5x5 averaging filter F_2 is able to boost the SNR by 5-10 dB. In high frequency bands ($\lambda^{-1} \approx 0.5$), the nonlinear grid filter is able to exploit its prior knowledge of what text should look like to boost the SNR by up to 25 dB.

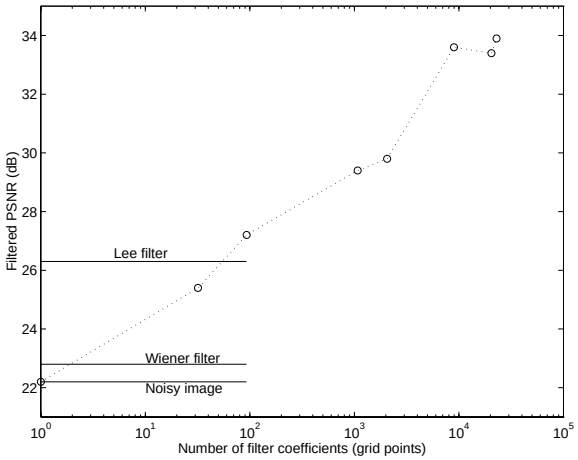


Figure 4: Filter performance as a function of filter size

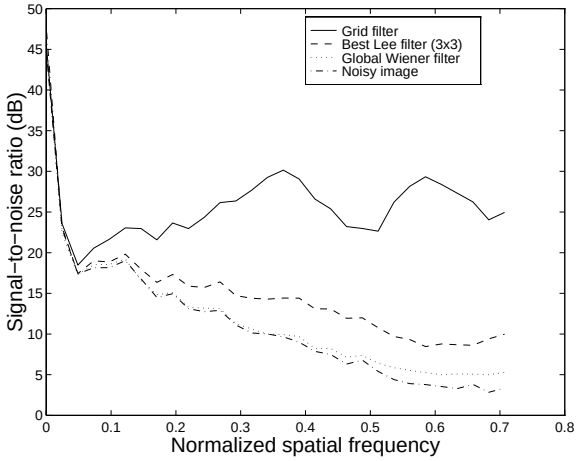
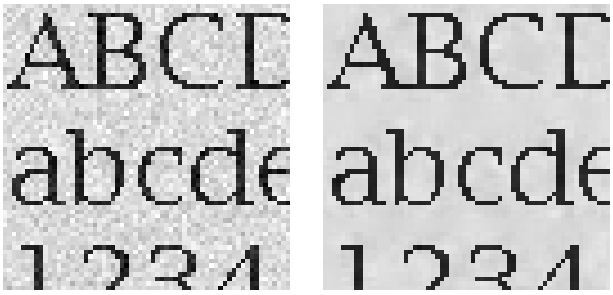


Figure 5: Signal to noise ratio

3.2. Faces with additive noise

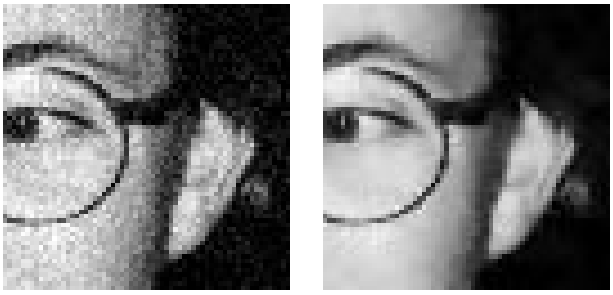
A filter using an 11^8 grid filter for F_1 and a 5x5 average for F_2 was trained on images of faces degraded by AWGN with $\sigma^2 = 200$. None of the faces in the training set were wearing glasses. The filter was tested on an image of a person wearing glasses (Figure 7). This image was 212x228 and required 10 seconds to filter. The strong, dark rim of the glasses was unlike anything present in the training sets, but was passed unchanged. This behaviour is un-



Noisy
MSE=400, PSNR=22.2

Filtered
MSE=27.5, PSNR=33.8

Figure 6: Restoration of 24-pt Palatino text using an 8^8 grid filter



Noisy
MSE=200 PSNR=25.2 dB

Filtered
MSE=46.8 PSNR=31.4 dB

Figure 7: Restoration of a face image using an 11^8 grid filter

like polynomial or feedforward network filters, which tend to have unpredictable behaviour for unexpected inputs.

3.2.1. Text with blurring and noise

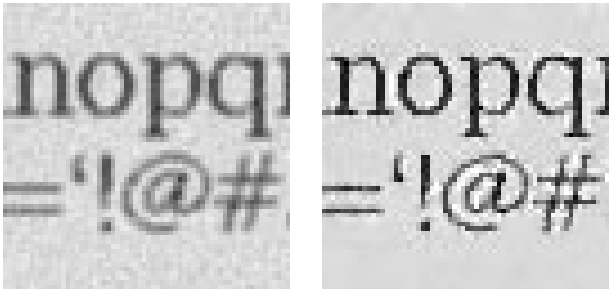
A filter using a 14^8 grid filter for F_1 and a 5x5 averager for F_2 was trained on images of text (Helvetica, Times-Roman, and Courier at 18 pt and 36 pt) degraded by blurring (a 3x3 mask) and AWGN with $\sigma^2 = 100$. Figure 8 shows an excerpt from a restoration of 24 pt Palatino text (not in the training set) using this filter. The image was 101x526 and required 9 seconds to filter. The grid filter is able to exploit the tightly-constrained nature of text images to simultaneously suppress noise and deblur the text.

4. LIMITATIONS

Grid filters are limited to a small number of inputs (roughly 13 if a typical workstation is used for training). We have had success with footprints as large as 15x15, with feature selection used to reduce the number of inputs.

Training grid filters requires pairs of pristine and degraded images. These training pairs can be acquired experimentally (for example, pairing low-quality images with those acquired by a high-fidelity imaging system). If the degradation process is well understood, degraded images can be simulated using appropriate synthetic images. Grid filters are not suitable for blind deconvolution.

Grid filters are good at removing local degradations, for example small amounts of blurring and white noise. They are not suitable for removing large-scale blurring or low-frequency noise,



Blurred and noisy
MSE=1869 PSNR=15.4 dB

Filtered
MSE=665 PSNR=19.9 dB

Figure 8: Restoration of blurred and noisy text using a 14^8 filter

although they might prove useful for cleaning up after a global inverse filter.

They perform best when the image class and degradation is tightly constrained. For example, a filter trained on document images with a specific amount of noise will perform much better than a filter trained to handle a wide variety of image types and degradations.

5. CONCLUDING REMARKS

We have described a new approach to local nonlinear image restoration based on the idea of approximating a nonlinear restoration function F on a grid of points in a many-dimensional space. Grid filters have $O(1)$ training and filtering complexity per pixel, independent of the number of filter coefficients. They are fast to train, numerically stable, leave unusual image features unchanged, and are a superset of order statistic filters.

Grid filters are best at undoing local degradations, for example small amounts of blurring and white noise. The more tightly the image class and degradation can be constrained, the better the resulting restoration.

6. REFERENCES

[1] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*. SIAM, Philadelphia, PA, 1994.

[2] A. C. Bovik, T. S. Huang, and D. C. Munson. A generalization of median filtering using linear combinations of order statistics. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 31(6):1342–1349, December 1983.

[3] I. Craig and J.C.Brown. *Inverse problems in astronomy: a guide to inversion strategies for remotely sensed data*. Adam Hilger Ltd., Boston, 1986.

[4] H. A. David. *Order statistics*. Wiley, Toronto, 1981.

[5] J. S. Lee. Digital image enhancement and noise filtering by use of local statistics. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, PAMI-2:165–168, 1980.

[6] J. Wood. Invariant pattern recognition: A review. *Pattern Recognition*, 29(1):1–17, 1996.