Title
Fast and stable Bayesian image expansion using sparse edge priors

Journal
IEEE Transactions on Image Processing, 16(4)

ISSN
1057-7149

Authors
Raj, Ashish
Thakur, Kailash

Publication Date
2007-04-01

Peer reviewed

Fast and Stable Bayesian Image Expansion Using Sparse Edge Priors

Ashish Raj and Kailash Thakur

***Abstract*—Smoothness assumptions in traditional image expansion cause blurring of edges and other high-frequency content that can be perceptually disturbing. Previous edge-preserving approaches are either ad hoc, statistically untenable, or computationally unattractive. We propose a new edge-driven stochastic prior image model and obtain the maximum *a posteriori* (MAP) estimate under this model. The MAP estimate is computationally challenging since it involves the inversion of very large matrices. An efficient algorithm is presented for expansion by dyadic factors. The technique exploits diagonalization of convolutional operators under the Fourier transform, and the sparsity of our edge prior, to speed up processing. Visual and quantitative comparison of our technique with other popular methods demonstrates its potential and promise.**

***Index Terms*—Bayesian estimation, edge-driven priors, image expansion, interpolation, subspace separation.**

I. Introduction

IMAGE expansion, whether by polynomial/spline interpolation or by more recent model-based methods, is an important, but evolving, field. This paper presents a Bayesian approach under a novel edge-driven prior image model and a realistic observation model. A novel efficient algorithm is then presented for fast processing of the computationally challenging Bayesian problem.

This paper is organized as follows. The rest of this section discusses the prior art and formulates the expansion problem. Section II proposes the edge-driven prior model, derives the Bayesian estimate, and describes our practical implementation. Section III develops an efficient subspace separation algorithm. Sections IV and V contain results and conclusions.

In our notation, scalar quantities are italicised (e.g., "x"), vectors are straight boldface ("x"), and matrices are upper case ("X"). X^T denotes the matrix transpose and X^H the conjugate transpose. The (i,j)th element of X is denoted by $X_{i,j}$. $Pr(\mathrm{x})$ denotes the joint probability distribution of the elements of vector x, and $E(\mathrm{x})$ denotes the expectation of x. We use $\mathrm{diag}(\mathrm{x})$ to denote a diagonal matrix whose diagonal entries are given by the vector x.

Manuscript received March 3, 2006; revised September 7, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Thierry Blu.

A. Raj is with the Center for Imaging of Neurodegenerative Diseases (CIND), University of California at San Francisco, VA Medical Center (114M), San Francisco, CA 94121 USA (e-mail: ashish.raj@ucsf.edu).

K. Thakur is with the Industrial Research Limited, Wellington, New Zealand (e-mail: k.thakur@irl.cri.nz).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2006.891339

A. Review

Images have traditionally been expanded by interpolating intermediate data points from the available coarse grid [1]. Piecewise polynomial fitting [2], splines [3], or convolution kernels [4] have been used for this purpose. Generalised interpolation was recently proposed using an approximation theory formulation [3], [5]. These methods have proved adequate for interpolation within the Nyquist rate, and for rotational or fractional resampling. However, there are two major problems with using these methods for image expansion by multiple factors. First, as described below, image expansion is not really the same as interpolation. Second, interpolation based on polynomial fitting, splines or convolution assumes global continuity and smoothness constraints that produce perceptually unsatisfactory results with blurred edges and textures.

Many *ad hoc* edge-preserving interpolation techniques have been reported to handle the second problem, including adaptive splines [6], POCS interpolation [7], nonlinear interpolation with edge fitting [8], and edge-directed interpolation [9]. The latter uses edge orientation to perform directional polynomial interpolation so that interpolation occurs only along an edge, and not across it. Problems include the need for a separate, accurate edge detection step and high computational complexity due to its use of iterative projections. The same is true of [7], another POCS method. In [8], a nonlinear interpolation is proposed that incorporates local edge fitting in order to avoid interpolation across edges. Space-warped polynomial interpolation [10] reduces edge blurring by assigning smaller weight to data samples on different sides of an edge than those on the same side. A rough local-gradient edge measure was proposed, but the method is highly sensitive to edge localization. The parameter choices are ad hoc and not locally adaptable. *Ad hoc* methods are nonoptimal since they do not use any underlying prior or likelihood model.

Some principled model-based approaches have also been reported. Wiener [11], Kalman [12], and adaptive-filter-based [13] methods achieve spatial adaptation, but at the cost of processing time. However, the value of these essentially causal/temporal techniques is unproven on images, whose statistics are liable to change abruptly and noncausally. Local statistic estimation requires large windows for accuracy but small windows for local adaptation, resulting in a limiting tradeoff. For instance, [11] proposed a promising way to interpolate between four neighboring pixels via a Wiener process to achieve sensitivity to edge orientation. Their results are impressive but do not indicate sufficient sensitivity to edges due to this tradeoff. Further, covariance estimates were obtained from the available low-resolution

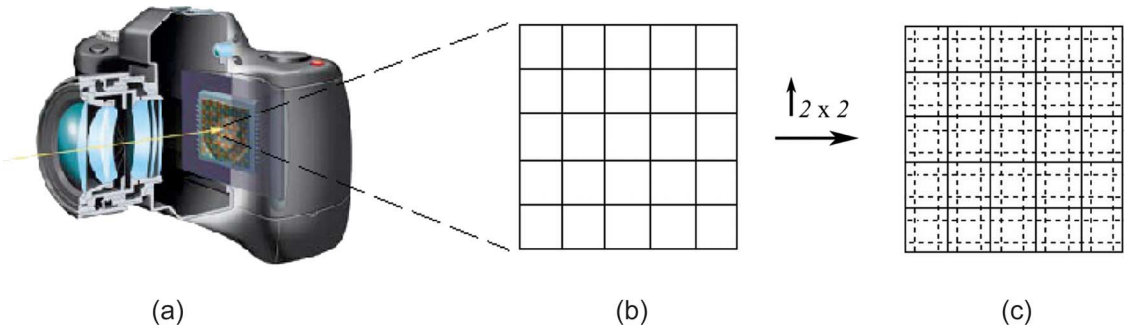

Fig. 1. (Solid lines) Camera acquires a low-resolution image by integrating over coarse pixels. (Dashed lines) The expansion task is to deduce intensity values that would arise if the integration is carried over a smaller pixel area, denoted by the finer grid.

image instead of the required (but not available) high-resolution image, a classic image expansion issue which can be neglected only for asymptotically dense sampling. An improvement to this technique was reported by Muresan *et al.* [14] who propose a novel technique, called AQua, for arbitrary expansion factors. AQua constrains the signal to belong to an adaptive quadratic signal class and reportedly shows better sharpness and edge preservation compared to [11]. However, both methods appear to exhibit excessive directional smoothing (see Section IV).

In order to obtain statistically sound expansion from under-sampled data, Bayesian methods using Markov random fields (MRFs) have been proposed [15]. Other Bayesian methods were reported by many authors using wavelets [16], EM algorithm [17]; an efficient algorithm was reported in [18]. Unfortunately, finding reliable estimates of prior distributions is difficult from available low-resolution data, and optimization of the resulting nonconvex cost functions is computationally intractable, even in simpler MRF techniques [15]. The Bayesian methods above have concentrated mainly on super-resolution from multiple low-resolution images. The expansion problem, in which only a single low-resolution image is available, is obviously more challenging. For this reason, fully Bayesian methods for arbitrary expansion factors using MRF priors appear too ambitious in terms of underlying models and computational feasibility.

In summary, current edge-preserving expansion methods are limited by two major issues.

1) They rely on accurate edge information that is, in fact, not obtainable from coarse data.
2) After incorporating edges, optimal Bayesian solutions are no longer computationally tractable.

In this paper, we propose a method to exploit sparsity in the prior model to achieve speed-up while retaining the essential characteristics of a locally variable Bayesian image model.

B. Problem Formulation

The problem we address is depicted in Fig. 1. The original "world" image intensity $x(t_1, t_2)$ undergoes blurring by the camera's optics, then spatially averaged over the sensor pixel area to produce a digital intensity value $x(j,k)$. Instead of trying to reproduce the exact continuous intensity $x(t_1, t_2)$ from the available discrete image $x(j,k)$, following [19], [20], in this paper, we consider instead a more realistic goal. Given intensity samples obtained at the coarse grid

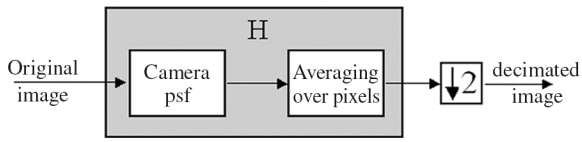

Fig. 2. Decimation model corresponding to (2). The effect of optical PSF and pixel averaging is combined in H.

$$x(j,k) = \int_{jT-T/2,kT-T/2}^{jT+T/2,kT+T/2} x(\tau_1,\tau_2)d\tau_1 d\tau_2 \qquad j \in \{1,2,\ldots N\}, k \in \{1,2,\ldots M\}$$

we wish to obtain the set of more finely sampled intensity values at the finer grid, assuming no aliasing on either grid. Fig. 1(b) and (c) show the two grids and how they are related to each other. Note the departure from traditional interpolation which assumes direct subsampling.

Approximating intensity to be uniform over each $T/2 \times T/2$ subpixel, denoting by the vector x the lexicographically arranged set of finely sampled intensities $\{x(j',k')\}$, and by y the coarser samples $\{x(j,k)\}$, this becomes a 2-D convolution followed by $2\times$ decimation (see Fig. 2)

$$\mathrm{y} = \downarrow_2 (\mathrm{h} \star\star \mathrm{x}) \tag{1}$$

with $\mathrm{h} = (1/16)\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$. Hence, image expansion involves not only up-sampling but also deconvolution. Camera PSF is included in h, if necessary, yielding finally the decimation model

$$\mathrm{y} = H\mathrm{x} + \mathrm{n} \tag{2}$$

where H is $(N/4 \times N)$, and n is additive noise, generally assumed i.i.d. white Gaussian. The decimation model (2) has been widely used in the problem of super-resolving satellite imagery from multiple low-resolution images [15]–[19]. It is related to a more general observation model described by [20], whose model simplifies to (2) under the assumption of negligible aliasing in the high-resolution image x. For expansion

factors 2^p, $p > 1$, we have

$$\mathrm{y} = [(\downarrow_2 \mathrm{h}) \star\star (\downarrow_2 \mathrm{h}) \star\star \cdots (\mathrm{p\ times})] \star\star \mathrm{x} + \mathrm{n} \triangleq H^{(p)}\mathrm{x} + \mathrm{n}$$

but we will solve this as a series of p $2\times$ interpolations, ignoring the fact that noise properties actually depend on p instead of being white Gaussian for all p. Our implementation focuses on dyadic expansion factors, but we note that this is a practical rather than theoretical restriction. In principle, a similar approach can be derived for any integer expansion factor.

Equation (2) can be solved directly via least squares methods traditionally used in decovolution/restoration [21], [22]. The expansion problem (2) is similar, but is, in addition, severely underdetermined. The least-squares methods can handle ill-posedness by introducing a regularization term $\mathcal{D}(\mathrm{x})$ which penalizes deviation from some *a priori* constraint. In Bayesian terms, this might represent knowledge of image covariance. Standard Tikhonov regularization employs global smoothness constraints: $\mathcal{D}(\mathrm{x}) = \|\mathcal{D}\mathrm{x}\|^2$, where matrix $\mathcal{D}$ represents the finite differences operator. The stabilizing term reduces noise amplification by preventing small singular values, acting as a filter on the matrix spectrum.

Unfortunately, Tikhonov-regularized least squares does not preserve the integrity of sharp edges due to unrealistic global smoothness assumptions on images, which are, in fact, better modeled as nonstationary processes. We now introduce an interesting edge-driven prior model of images, and derive the optimal *maximum a posteriori* (MAP) estimate for expansion.

C. Our Approach

MAP methods are computationally prohibitive since matrix diagonalization is no longer possible by Fourier transformation. We will model both the likelihood and the prior in terms of L_2 distances, which makes the MAP problem essentially quadratic. We reduce complexity by exploiting sparsity of discontinuous transitions and obtain a "near-diagonalization" result which allows fast processing. Quadratic inverses involving variable weights typically suffer from instability and ill-conditioning. Our approach divides the solution space into two orthogonal subspaces, which allows the quadratic inverse to be obtained independently and stably within each subspace.

For a practical implementation, we further reduce computational burden by tiling the image and solving a series of 1-D problems, first on rows, followed by columns. This is possible if the decimation kernel is separable into row and column vectors. The method is naturally progressive—simpler, approximate estimates are obtained first, and fed back to obtain more exact (and expensive) estimates. Our derivations assume that a large sparse Toeplitz matrix can be replaced by a circulant one without error—a standard assumption in many Fourier methods. In images, this assumption causes problems only near boundaries, and is easily mitigated by zero padding.

II. Maximum *A Posteriori* (MAP) Image Expansion Using Prior Edge Information

A. Sparse Edge-Driven Prior Model

Images are well modeled by piecewise smoothness punctuated with edge discontinuities [23]. Traditionally, images have been stochastically modeled as having translation-invariant global smoothness of the kind $\mathrm{x} = A\mathrm{p}$, where p is an i.i.d. signal and matrix A models the spatial correlation of the image signal and is usually expressed as a Toeplitz matrix corresponding to some smoothing convolution kernel. It has been shown that under this model MAP reproduces Tikhonov regularization with the equivalence $\mathcal{D} = A^{-1}$ [24]. We wish to modify this model to incorporate edge priors. The classic difficulty of this approach is that the information about edge location, orientation and magnitude is data-dependent; hence, it can hardly be called a "prior" in the normal sense. Therefore, direct use of edge detection results as edge priors may be considered statistically unsound under noise, blur and other artifacts. Certainly, this is highly problematic for image expansion where the data are highly blurred and decimated, although limited justification for this approach was presented in [11]. In this paper, we argue that although accurate edge priors relying on steepness and resolution of edges in x cannot be obtained from available data y, the *approximate* location and magnitude of edges can be easily and reliably obtained. Instead of using edge information as a hard decision step, we propose a stochastic framework to allow soft decisions.

For the purpose of edge detection, we first expand the observed image y to the same size as desired image x, and obtain $\overline{\mathrm{y}}$. We used bilinear interpolation for this, but any popular method could be used, for instance polynomial kernel or nearest neighbor. Let us denote by $\mathcal{E}_y = \{j \mid \overline{y}_j \text{ is an edge in image } \overline{\mathrm{y}}\}$ the set of edge locations obtained from edge detection on $\overline{\mathrm{y}}$. Edge set $\mathcal{E}_y$ is obtained as follows; the coarse image $\overline{\mathrm{y}}$ is filtered by the derivative of Gaussian (DoG) to get a rough map of intensity slopes [25]. The width of DoG was empirically chosen to match the noise. A small width allows finer edges to be registered, but increases false detection due to noise. Some amount of initial operator supervision is, therefore, inevitable. We do not implement complex and expensive edge detection steps or cross validation for optimal edge parameters since our aim is to minimize execution time. Simple edge detection followed by hierarchical edge linking [26] was found sufficient. Image $\overline{\mathrm{y}}$ is a blurry, low-resolution image, so more sophisticated edge detection methods are not helpful; in any case, our stochastic model shown below serves to desensitize the effect of edge detection.

We introduce a sparse, independent "edge" point process $\mathrm{b} = \chi \cdot \mathrm{q}$, where χ is an independent binomial point process which can take the values 0 or 1, q is an independent Gaussian stochastic process, and multiplication "$\cdot$" is point-by-point. Our probability model is

$$\begin{aligned} Pr(\mathrm{b}) &= \prod_i Pr(\chi_i) Pr(q_i) \\ Pr(\chi_i = 1) &= \frac{1}{\mathrm{const}} \sum_j \phi(i), \quad j \in \mathcal{E}_y \\ Pr(\chi_i = 0) &= 1 - Pr(\chi_i = 1) \\ q_i &\sim \mathcal{N}\left(0, \sigma_b^2\right) \end{aligned}$$

where σ_b^2 is the variance of identically distributed Gaussians q_i and ϕ is an inverse distance specified later. Let p be an i.i.d.

stochastic process with variance σ_x^2[1] and A be a Toeplitz matrix that captures the spatial correlation of the image. Then our new image model is

$$\mathrm{x} = A(\mathrm{p} + \mathrm{b}). \tag{3}$$

This reproduces a spatially correlated (smooth) image process interspersed with edge point processes of random but Gaussian distributed magnitude. This edge model may not be entirely satisfactory for 2-D images, but is justifiable under our separable 1-D approach.

We now make the standard assumption: Process x is jointly (zero mean) Gaussian, with

$$Pr(\mathrm{x}) = \exp\left\{-\frac{1}{2}\mathrm{x}^T R_x^{-1}\mathrm{x}\right\} \tag{4}$$

where $R_x \triangleq E(\mathrm{x}\mathrm{x}^T)$ is the signal covariance matrix. From (3), we have $E(\mathrm{x}\mathrm{x}^T) = A(E(\mathrm{p}\mathrm{p}^T) + E(\mathrm{b}\mathrm{b}^T))A^T$, assuming independence between p and b. Using the independence of p and b, $E(\mathrm{p}\mathrm{p}^T) = \sigma_x^2 I$ and $E(\mathrm{b}\mathrm{b}^T) = \mathrm{diag}(E(b_i^2))$. From (3)

$$E\left(b_i^2\right) = Pr(\chi_i = 1)\cdot E\left(q_i^2\right) = \frac{\sigma_b^2}{\mathrm{const}}\sum_j \phi(i)$$

$$R_x = \sigma_x^2 A W^2 A^T.$$

The diagonal matrix W is given by $W_{i,i} = \sqrt{1 + c\sum_j \phi(i)}$, $j \in \mathcal{E}_y$, and $c = \sigma_b^2/(\mathrm{const}\cdot\sigma_x^2)$. For convenience we define an edge-weighted map $B_{\mathcal{E}}(i) \triangleq W_{i,i}^{-1} = 1/\sqrt{1 + c\sum_j \phi(i)}$, $j \in \mathcal{E}_y$.

Since the appropriate form of ϕ is not known *a priori*, we have tried various intuitive forms including Gaussian, linear, quadratic, etc. Overall, the inverted truncated linear model $\phi(i) = (1/\Delta)\max(\Delta - |i-j|, 0)$ seemed to work best—this appears to mirror current literature on robust image estimation [18], [27]. For this choice of ϕ, $B_{\mathcal{E}}$ forms a distance metric, and can be efficiently computed in terms of a distance transform in linear time [28]. Note that σ_b and const are prior model parameters which are basically unknown, and such quantities are determined heuristically after trial and error in most Bayesian applications. We found the best results by setting $c = 9$ and $\Delta = 3$. The plot of $\phi(i)$ for these settings and the corresponding values of weights $W_{i,i}$ and $B_{\mathcal{E}}(i)$ are shown in Fig. 3 as a function of i in the vicinity of an edge location $j \in \mathcal{E}_y$. Adaptive and spatially variable choice of these parameters can lead to improved performance, but this is a subject of future work. Note also that, in practice, the elements of $\mathcal{E}_y$ are so far apart that it is possible to use nearest neighbor processing to speed up the computation of the distance transform.

B. Deriving the MAP Estimate

Using this prior, we wish to estimate x. The optimal MAP estimate maximizes the posterior $Pr(\mathrm{x}|\mathrm{y}) \propto Pr(\mathrm{y}|\mathrm{x})Pr(\mathrm{x})$. The likelihood function $Pr(\mathrm{y}|\mathrm{x})$ comes from the decimation model

$$Pr(\mathrm{y}|\mathrm{x}) = \exp\left\{-\frac{1}{2\sigma_n^2}\|\mathrm{y} - H\mathrm{x}\|^2\right\}. \tag{5}$$

MAP is typically obtained by minimizing $-\log(Pr(\mathrm{x}|\mathrm{y}))$, which from (4) and (5) becomes

$$\mathrm{x}_{\mathrm{MAP}} = \arg\min_{\mathrm{x}} \|\mathrm{y} - H\mathrm{x}\|^2 + \mu^2 \mathrm{x}^T R_x^{-1}\mathrm{x}. \tag{6}$$

The parameter μ is theoretically determined by prior and noise energy: $\mu = \sigma_n/\sigma_x$. In practice, good unsupervised estimates of these variances might be difficult, so μ is chosen empirically. Many cross-validation techniques are available to automate the process [29], but this issue was not explored in the current work. The closed form solution of (6) using $\mathcal{D} = A^{-1}$ is given by

$$G = \mu W^{-1}\mathcal{D}$$

$$\mathrm{x}_{\mathrm{MAP}} = (H^T H + G^T G)^{-1} H^T \mathrm{y} \triangleq H^{\dagger}\mathrm{y}. \tag{7}$$

Here, we have defined the "pseudo-inverse" $H^{\dagger}$ as the regularized inverse of H. This closely resembles a regularized least square problem but with variable regularization weights. The MAP estimate basically imposes a locally varying smoothness constraint whose local weights come from prior knowledge of edge locations $j \in \mathcal{E}_y$ and the assumed distance function $\phi(\cdot)$. This allows for a probabilistic rather than hard interpretation of edge information deduced from observation y, thus providing a more robust and statistically sound approach.

A direct matrix inversion of (7) is not advisable due to prohibitive computational cost, as well as instability and ill-conditioning in regions with small weights. Section III proposes instead a new computationally efficient and stable subspace-based inversion algorithm. However, first we give an overview of the image expansion algorithm.

C. Practical Implementation Using Separable 1-D Processing

To reduce computational burden, deconvolution is sometimes done one row at a time or over tiled subimages [30]. We take this approach by making the separable assumption on h.

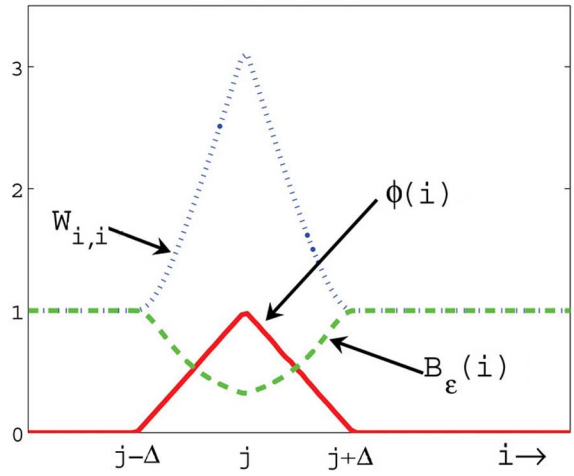

Fig. 3. Plot of edge-driven functions used as edge prior in this paper: (solid) ϕ, (dashed) $B_{\mathcal{E}}$ and (dotted) W. All curves are for i in vicinity of edge location j, assuming there are no other edge locations in this neighborhood.

[1]Subscript x emphasizes that process p captures the statistics of image x, including its variance

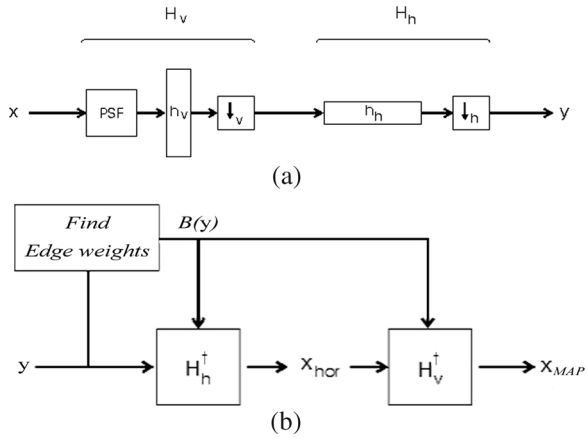

Fig. 4. (a) Separable decimation model showing how a low-resolution image y arises from x via separate horizontal and vertical processes. The convolution kernels h_v and h_h are 1-D. (b) The Separable MAP solution is obtained by "undoing" the horizontal and vertical decimations in (a) sequentially. Operations denoted by $H_h^\dagger$ and $H_v^\dagger$ are described in Section III.

Definition: The 2-D impulse response h is considered *separable* if it can be expressed as

$$\mathrm{h} = \mathrm{h}_v \mathrm{h}_h^T$$

where h_v and h_h are column vectors representing vertical and horizontal impulse response. This assumption is clearly valid for our model h, with $\mathrm{h}_v = 1/4[1,2,1]^T$ and $\mathrm{h}_h = 1/4[1,2,1]$. If separability does not hold, Fig. 2 can still be resolved as shown in Fig. 4(a).

Our separable scheme first inverts horizontal decimation, then uses the result to invert vertical decimation, as shown in Fig. 4(b). Due to separability, each 1-D inversion $H_h^\dagger$ and $H_v^\dagger$ is performed independently on rows and columns of y, via an efficient subspace method (Section III). To further reduce computational burden, we break the image into several subimages, and perform the separable operations on each. In practice, the subimage grid is designed so that there is some amount of overlap between adjacent subimages. The overlap allows a better merging of subimages without causing grid artifacts. The overall algorithm is presented as follows.

Algorithm A1: Overall Separable Image Expansion Algorithm

- Compute weight map $B_\mathcal{E}$ and diagonal weight matrix W as in Section II-A.
- Break y into subimages y_s^k, $k = 1 \ldots K$, of size $(m \times n)$. Similarly break $B_\mathcal{E}$ into ${B_\mathcal{E}}_s^k$.
- For each subimage $k = 1 \ldots K$.
  1) Compute the inversion $\mathrm{x}_{\mathrm{hor}} = H_h^\dagger y_s^k$, as follows:
     — For $i = 1 \ldots m$
     $$(i\text{th row of } \mathrm{x}_{\mathrm{hor}})^T = H_h^\dagger \cdot \left(i\text{th row of } y_s^k\right)^T$$
     where inversion $H_h^\dagger$ is via algorithm A2 in Section III, and uses the weight map ${B_\mathcal{E}}_s^k$.
  2) $\mathrm{x}_{\mathrm{hor}}$ is now of size $(m \times 2n)$.
  3) Compute the inversion $\mathrm{x}_{\mathrm{est}} = H_v^\dagger \mathrm{x}_{\mathrm{hor}}$, as follows:
     — For $j = 1 \ldots 2n$
     $$(j\text{th column of } \mathrm{x}_{\mathrm{est}}) = H_v^\dagger \cdot (j\text{th column of } \mathrm{x}_{\mathrm{hor}})$$
     where inversion $H_v^\dagger$ is via algorithm A2.
  4) Assign $x_s^k = \mathrm{x}_{\mathrm{est}}$ as the kth expanded subimage.
- Stitch all x_s^ks back into the overall expanded image x.
- If expansion factor is 2^p, $p > 1$, then repeat the whole process p times.

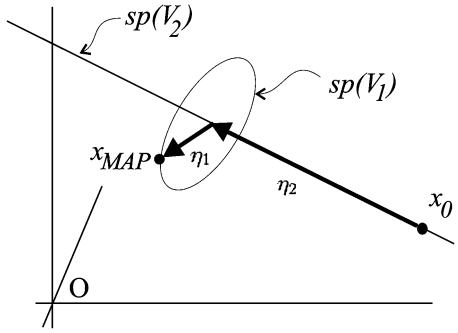

Fig. 5. Pictorial representation of subspace partitioning—searching for cost minimization staring from x_0, and ending at $\mathrm{x}_{\mathrm{MAP}}$.

III. New Subspace Separation Method for MAP Image Expansion

Direct inversion of (7) suffers from high computational cost as well as instability and noise amplification in regions with small weights. To address the computational issue, many papers on deconvolution and restoration have previously proposed diagonalizing of least squares problems like (7) via the DFT. Unfortunately, due to variable weights W in the MAP formulation, this is no longer possible. A new stable and efficient framework is now obtained by basis change which "almost diagonalizes" the problem. Solution space (spanned by some unitary matrix V) is partitioned into two orthogonal subspaces $sp(V_1)$ and $sp(V_2)$, so that the inverse problem is independent, well-posed and stable in each subspace

$$\mathrm{x}_{\mathrm{MAP}} = \mathrm{x}_0 + V_1 \eta_1 + V_2 \eta_2 \tag{8}$$

where x_0 is any solution to the under-determined system $H\mathrm{x} = \mathrm{y}$, and η_1, η_2 are vectors of length $N/2$. Fig. 5 depicts the situation, where x_0 can be anywhere on the line defining $sp(V_2)$, and $sp(V_1)$ is shown orthogonal to $sp(V_2)$. V_1 and V_2 are given by the SVD of H.

Henceforth, the minimization of (6) is conducted over (η_1, η_2). We will show that this basis change "almost bidiagonalizes" the system and in the absence of variable weights solves a bidiagonal system. To efficiently handle variable weights in the MAP estimate, we exploit the sparse edge assumption. These three steps, depicted in Fig. 6 are now explained in detail.

1) Specifying V_1, V_2, x_0: SVD of Toeplitz matrices is given by the Fourier Transform. It is well-known [31] that the following approximate Toeplitz diagonalization applies

$$H_{\mathrm{full}} \approx D_N \Sigma D_N^H \tag{9}$$

where D_N is the unitary N-point DFT matrix and diagonal $\Sigma = \mathrm{diag}(\{\sigma_k\})$ contains the singular values of H_{full}, which are in turn given by the DFT of the impulse response $\mathrm{h}_{\mathrm{full}}$. Equation (9) is accurate when the length of $\mathrm{h}_{\mathrm{full}}$ is much smaller than N, and becomes perfect in the asymptote. Potential errors due to

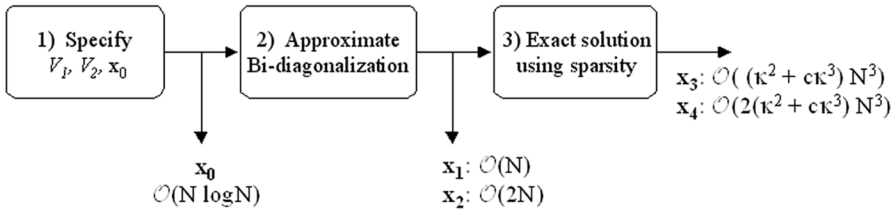

Fig. 6. Three steps involved in finding the MAP estimate. Estimates x_0 to x_3 are intermediate solutions, and x_4 is the exact MAP solution. These quantities are summarized under Algorithm A2. The *incremental* numerical cost of each estimate from the preceding step is shown alongside. κ is the sparsity ratio of the edge weights.

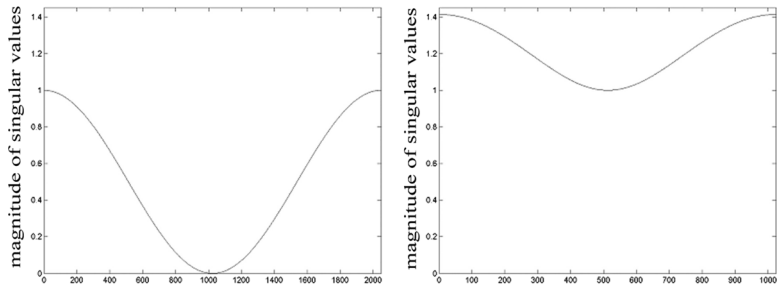

Fig. 7. (a) Singular values of H_{full} (i.e., diagonal elements of Σ), and (b) of decimated matrix H (i.e., diagonal elements of Σ_H).

this approximation occur only near image boundaries, and are easily mitigated by zero padding.

Analogous results for the row decimated matrix H appear to have gone unreported, which we now explicitly derive. Recall that skipping every other point of a signal amounts to aliasing (wrapping) in the Fourier domain. H is obtained from H_{full} by skipping rows; hence, the singular values of H must come from the aliasing of the frequency response of h_{full}. Define permutation operators P_e and P_o which select the even and odd rows, respectively. Define also the overall permutation $P = [P_e^T, P_o^T]^T$. Then, the SVD of H is summarized in Lemma 1, with proof in Appendix A.

Lemma 1: Let $D_{N/2}$ be the $N/2$-point DFT matrix and E a diagonal phase matrix: $E = \text{diag}(\{e^{-j(2\pi/N)k}\}_{k=0\ldots N-1})$. Let $\Sigma = \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix}$. Then, the SVD of H is given by

$$\begin{aligned} H &= D_{N/2}(\Sigma_H \quad 0)V^H = D_{N/2}(\Sigma_H \quad 0)(V_1 \quad V_2)^H \\ &= D_{N/2}\Sigma_H V_1^H, \\ |\Sigma_H|^2 &= |\Sigma_1 + \Sigma_2|^2 + |\Sigma_1 - \Sigma_2|^2 \\ V_1 &= P^T \begin{pmatrix} D_{N/2}\Sigma_H^{-1}(\Sigma_1 + \Sigma_2) \\ D_{N/2}E\Sigma_H^{-1}(\Sigma_1 - \Sigma_2) \end{pmatrix} \\ V_2 &= P^T \begin{pmatrix} D_{N/2}\Sigma_H^{-1}(\Sigma_1 - \Sigma_2) \\ -D_{N/2}E\Sigma_H^{-1}(\Sigma_1 + \Sigma_2) \end{pmatrix}. \end{aligned}$$

Recall that h is real and symmetric, and the impulse response corresponding to $\mathcal{D}$ is assumed real and anti-symmetric. Hence, Σ_1, Σ_2, and Σ_H are all real, and $|\cdot|$ is not needed above.

Fig. 7 shows the singular values of a typical convolution matrix H_{full} (i.e., the diagonal elements of Σ) corresponding to $h = [0.25\ 0.5\ 0.25]$. Singular values of the associated decimated matrix H (i.e., the diagonal elements of Σ_H) are also plotted. We note that even though the full matrix has some small singular values, making the system badly conditioned, the decimated matrix possesses good conditioning, which guarantees that Σ_H^{-1} exists and is bounded.

For x_0, any vector satisfying $Hx_0 = y$ will do, but we select the one with minimum norm

$$x_0 = H^T(HH^T)^{-1}y = V_1\Sigma_H^{-1}D_{N/2}^H y. \tag{10}$$

It is easy to show that both the expression in the middle above ("right pseudo-inverse") and on the right ("truncated SVD") are equal, have the minimum norm, and are computed in $\mathcal{O}(N \log N)$ time. In comparison the traditional way of doing direct matrix inversion in (7) via the SVD would take $\mathcal{O}(N^3)$ flops in general.

2) Approximate Bidiagonalization of (7): With a change of basis, (6) becomes

$$\begin{aligned} \eta_1, \eta_2 = \arg\min_{\eta_1,\eta_2} \Big\{ &\|y - H(V_1\eta_1 + V_2\eta_2 + x_0)\|^2 \\ &+ \|G(V_1\eta_1 + V_2\eta_2 + x_0)\|^2 \Big\}. \end{aligned} \tag{11}$$

Taking partial derivatives and equating to zero, we get

$$\begin{aligned} V_1^H(S+Q)V_1\eta_1 + V_1^H(S+Q)V_2\eta_2 &= -V_1^HQx_0 \\ V_2^H(S+Q)V_1\eta_1 + V_2^H(S+Q)V_2\eta_2 &= -V_2^HQx_0 \end{aligned}$$

where $S = H^TH$, and $Q = G^TG$. In fact, the turning points are minima, since S and Q are positive definite. Using the SVD of H, we have $V_1^HSV_1 = \Sigma_H^2$, and $V_2^HSV_1 = V_1^HSV_2 = V_2^HSV_2 = 0$. Thus

$$\begin{pmatrix} \Sigma_H^2 + V_1^HQV_1 & V_1^HQV_2 \\ V_2^HQV_1 & V_2^HQV_2 \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} = -\begin{pmatrix} V_1^HQx_0 \\ V_2^HQx_0 \end{pmatrix}. \tag{12}$$

In absence of variable weights, G is diagonalized by the DFT because $\mathcal{D}$ is Toeplitz: $\mathcal{D} = D_N\Sigma_d D_N^H$. Since V_1 and V_2 both contain aliased DFT vectors, it is the case (see Appendix B) that

all matrices in (12) are, in fact, diagonal in the absence of W. Thus, (12) becomes a bidiagonal system and is speedily solved in linear time. However, variable weights destroy diagonalization, and (12) appears no better than the original (7). However, in practice, the variable weights are *sparse*, and approximate diagonalization continues to hold, due to Theorem 1.

Theorem 1: Let S, Q, V_1, V_2 be as before. Let $\Sigma = \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix}$ and $\Sigma_d = \begin{pmatrix} \Sigma_{d1} & \\ & \Sigma_{d2} \end{pmatrix}$. Let us assume that the diagonal entries of W are mostly one, except at a small proportion of points $\kappa \ll 1$. Then, the system (12) can be split into a "dominant" bidiagonal system $\Gamma = \begin{pmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{pmatrix}$ with diagonal Γ_{ij}, and a "residual" nonsparse $C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$. Specifically

$$(\Gamma + C)\begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} = \underline{\mathrm{y}} \tag{13}$$

and $\underline{\mathrm{y}} = -\begin{pmatrix} \Gamma_{11} + C_{11} \\ \Gamma_{21} + C_{21} \end{pmatrix} \Sigma_H^{-1} D_{N/2}^H \mathrm{y}$. C_{ij} satisfy Frobenius norm conditions $\|C_{ij}\|_F^2 < \kappa \|\Gamma_{ij}\|_F^2$.

Proof: Expressions for Γ_{ij} and proof are in Appendix B.■

Recall that for $(N \times N)$ matrix A, the Frobenius norm is $|A\|_F^2 \triangleq 1/N \sum_{i,j} |A_{ij}|^2$. Theorem 1 is a statement on the accuracy of the diagonalization of (12) in terms of Frobenius norm of nondiagonal matrix elements. It basically says that for sufficiently sparse weights, the MAP estimate is close to the globally smooth Tikhonov regularized estimate, and specifies a fast algorithm to obtain it in $\mathcal{O}(N \log N)$ time (by computing Γ).

On the other hand, the full, exact MAP solution is computationally expensive in general, requiring $\mathcal{O}(cN^3)$ operations, where c is usually much higher than unity. However, (12) is in fact strongly decoupled for small μ. To see this, consider the top block-row of (13). Due to the smallness of μ, we expect $\|\eta_1\| \ll \|\eta_2\|$. Thus, we propose a decoupled algorithm to solve (12).

1) Solve the bidiagonal system

$$\begin{pmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{pmatrix} \begin{pmatrix} \eta_1^0 \\ \eta_2^0 \end{pmatrix} = \underline{\mathrm{y}} \triangleq \begin{pmatrix} \underline{\mathrm{y}_1} \\ \underline{\mathrm{y}_2} \end{pmatrix}.$$

2) Solve for η_2

$$(\Gamma_{22} + C_{22})\eta_2 = \underline{\mathrm{y}_2} - (\Gamma_{21} + C_{21})\eta_1^0.$$

3) Finally, solve for η_1

$$(\Gamma_{11} + C_{11})\eta_1 = \underline{\mathrm{y}_1} - (\Gamma_{12} + C_{12})\eta_2.$$

Step 1) is linear time since it is a bidiagonal system. The errors due to the use of η_1^0 instead of η_1 is expected to be small since we expect $\|\eta_1\| \ll \|\eta_2\|$. Therefore , Step 2) is reasonable. Steps 2) and 3) could be iterated in an EM-type loop for greater accuracy, but this was not found to produce additional improvement. The MAP solution is given by

$$\mathrm{x}_{\mathrm{MAP}} = \mathrm{x}_0 + V_1\eta_1 + V_2\eta_2. \tag{14}$$

3) Efficient Inversion of (12) Using Sparsity of W: We now describe an efficient and stable algorithm for Step 2), involving inversion of $N/2 \times N/2$ diagonally dominant matrices. Step 3) uses the same algorithm. Following Appendix B, C_{22} is written as

$$C_{22} = \begin{pmatrix} \Lambda_2^1 & \Lambda_2^2 \end{pmatrix} D_N^H \overline{W}^2 D_N \begin{pmatrix} \Lambda_2^1 \\ \Lambda_2^2 \end{pmatrix}. \tag{15}$$

Since most entries of the diagonal matrix $\overline{W}$ are zero, C_{22} is rank deficient, which implies that $C_{22} = UU^H$, where U is a $(N \times \kappa N)$ matrix given by $U = (\Lambda_2^1 \ \Lambda_2^2)\widetilde{D_N^*}\widetilde{W}$. Here, we replaced $\overline{W}$ with the $(\kappa N \times \kappa N)$ matrix $\widetilde{W}$ consisting of only nonzero diagonal elements, and the $(N \times \kappa N)$ matrix $\widetilde{D_N}^*$ is obtained by removing the corresponding columns of D_N^H. Using the formula [32]

$$(A + UU^H)^{-1} = A^{-1} - A^{-1}U(U^H A^{-1} U + I)^{-1} U^H A^{-1} \tag{16}$$

and letting $A = \Gamma_{22}$ yields the inverse of $(\Gamma_{22} + C_{22})$. (16) allows us to invert a $(\kappa N \times \kappa N)$ matrix instead of a $(N \times N)$ matrix. Careful inspection will reveal that the total cost of computing this inverse is $\mathcal{O}(\kappa^2 N^3 + c\kappa^3 N^3)$. Typically, $\kappa < 0.2$, and the computational cost of inverting $(\Gamma_{22} + C_{22})$ is several orders of magnitude smaller than direct inversion. As summarized below, the subspace algorithm produces four approximate solutions that progressively build on each other. The numerical cost of each step was summarized in Fig. 6.

Algorithm A2: Progressive solutions to the full 1-D MAP Expansion problem

1) Find the minimum norm/pseudo-inverse/truncated SVD solution

$$\mathrm{x}_0 = V_1 \Sigma_H^{-1} D_{N/2}^H \mathrm{y}.$$

2) Solve the bidiagonal system

$$\begin{pmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{pmatrix} \begin{pmatrix} \eta_1^0 \\ \eta_2^0 \end{pmatrix} = \underline{\mathrm{y}} \triangleq \begin{pmatrix} \underline{\mathrm{y}_1} \\ \underline{\mathrm{y}_2} \end{pmatrix}.$$

$$\mathrm{x}_1 = \mathrm{x}_0 + V_2\eta_2^0$$

$$\mathrm{x}_2 = \mathrm{x}_0 + V_1\eta_1^0 + V_2\eta_2^0$$

3) Obtain η_2 as follows.
   (a) Create the "thin" $(N \times \kappa N)$ matrix U from the nonzero columns of $D_N^H W$.
   (b) Compute $U = (\Lambda_2^1 \ \Lambda_2^2)U$ by simple scaling and addition.
   (c) Replace $\eta_1^0 \leftarrow \Gamma_{22}^{-1}[\underline{y_2} - (\Gamma_{21} + C_{21})\eta_1^0]$ by simple matrix-vector multiply.
   (d) Compute the "small" matrix inverse $Q_{\mathrm{inv}} = (U^H \Gamma_{22}^{-1} U + I)^{-1}$.
   (e) Get $\eta_2 = \eta_1^0 - \Gamma_{22}^{-1} U Q_{\mathrm{inv}} U^H \eta_1^0$ by simple matrix-vector multiply.
4) Get η_1 by steps analogous to (a)–(e) above.
5) Then, finally

$$\mathrm{x}_3 = \mathrm{x}_0 + V_2\eta_2$$

$$\mathrm{x}_4 = \mathrm{x}_{\mathrm{MAP}} = \mathrm{x}_0 + V_1\eta_1 + V_2\eta_2$$

Estimates x_0, x_1, and x_2 are very quickly obtained in almost linear time, but only provide globally smooth results with

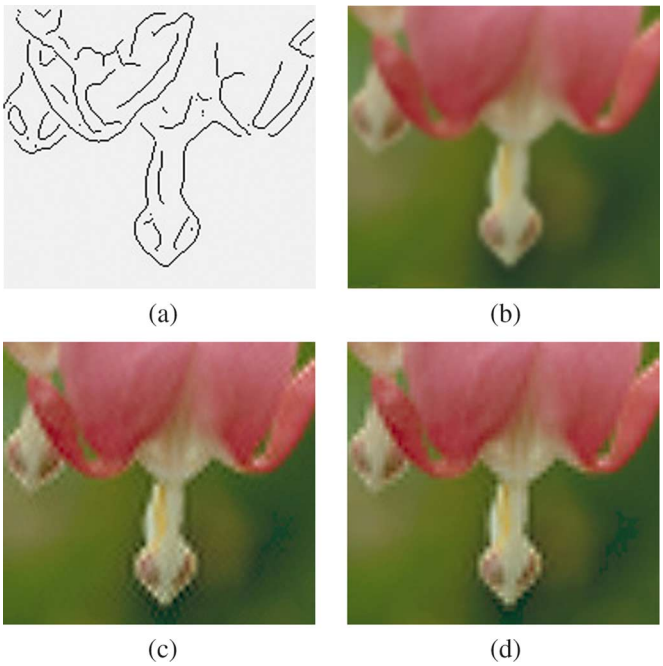

Fig. 8. Results of 4×4 expansion on a region of the "Flowers" image. (a) Edge map obtained from the low-resolution image, which is used to derive the sparse edge prior. (b) Cubic interpolated result. (c) Result of intermediate solution x_2. (d) MAP solution $\mathrm{x}_{\mathrm{MAP}}$.

ringing near sharp edges (see Fig. 8). Both x_3 and x_4 prevent blurring across edges, but at higher cost. x_3 is cheaper than x_4 (the full MAP solution) since it requires only one matrix inversion instead of two. x_3 is in fact a constrained least squares [21] solution of the MAP problem—see Appendix C.

IV. Results

Experiments were conducted on a set of high-resolution images as follows. The original image was decimated according to Section I-B and fed into our and other popular methods. First, we show the difference between the linear-time approximate solution (x_2 in Section III), and the more expensive MAP solution ($\mathrm{x}_{\mathrm{MAP}}$), in Fig. 8, which shows the $4\times$ expansion of the "Flowers" image. The edge map in (a) was obtained from the observed low-resolution image as described in Section II. The main difference between our two solutions is at or near object boundaries indicated by the edge map, where the MAP result is noticeably better.

Fig. 9 contains comparative results on the "Bike" image. A portion of the original image is shown in Fig. 9(a). In Fig. 9(b)–(e), we show results from the following algorithms.

1) Cubic—Keys' standard cubic interpolator [4]—Fig. 9(b).
2) AQua—Optimal recovery interpolation based on Golomb–Weinberger theory [14]—Fig. 9(c). AQua results were obtained from code supplied by the authors of [14].
3) AltaMira—A commercial Fractal-based image expansion from AltaMira Corp. [33]—Fig. 9(d). This undisclosed algorithm can be considered an industry benchmark, and has been widely used for comparative evaluation in recent literature [34], [35].
4) MAP—Our fast, sparse MAP image expansion—Fig. 9(e).

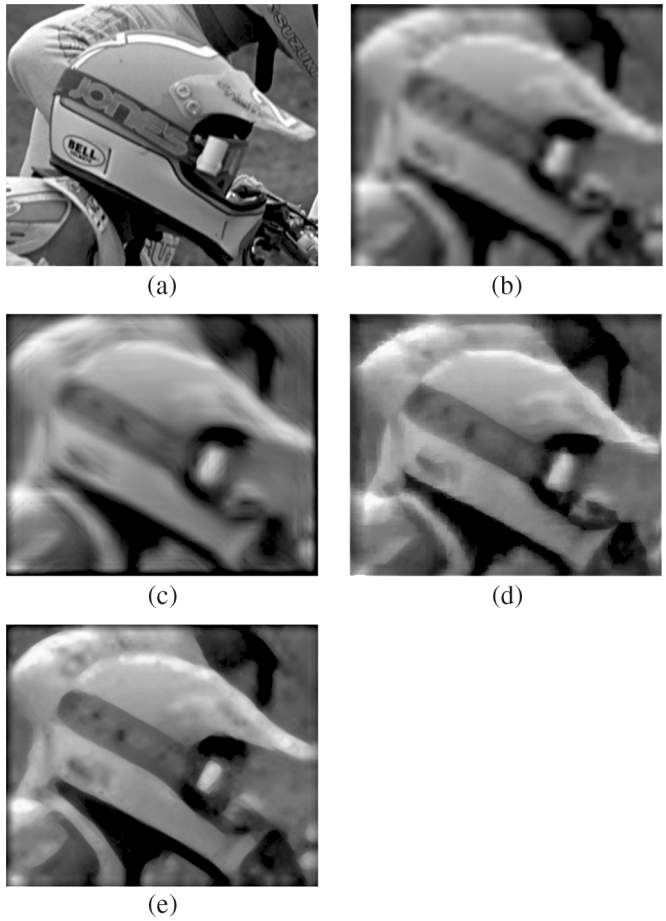

Fig. 9. Results of 8×8 expansion on a region of the "Bike" image. (a) Original.(b) Cubic. (c) AQua. (d) Altamira. (e) MAP.

This is repeated in Figs. 10 and 11 for the "Face" and "Lighthouse" images. We show 8×8 expansion results for the "Bike" image. The factor 8 represents an extreme case, and shows up many artifacts that crop up during expansion by various methods. The "Face" results are also 8×8, while the "Lighthouse" results are 2×2.

The following summarizes the parameters used and other relevant choices.

1) We used $\mathrm{h} = [0.25, 0.5, 0.25]$, and omitted camera PSF. If camera PSF is not negligible and not known, it will obviously affect performance. We have not investigated sensitivity to PSF mismatch, but is probably noncritical in comparison with pixel averaging. Experimentation with additional PSFs indicated little change in the relative performance of the algorithms.
2) Finite difference operator: We used the first difference kernel $\mathrm{d} = [-1, 1]/2$. Longer, smoother kernels were also tried, but without perceptible change.
3) The regularization parameter μ was determined in each case after a small number of trials; $\mu = 0.1$ yielded a reasonable result in most cases.
4) The size of tiling blocks was set at 64×64, with overlap margins of ten pixels on each side.

The results show the edge preserving effect of our MAP algorithm. Sharpness around edges, which was lost in both cubic and AQua interpolation, was recovered by MAP. The proprietary fractal-based software AltaMira does better than cubic or

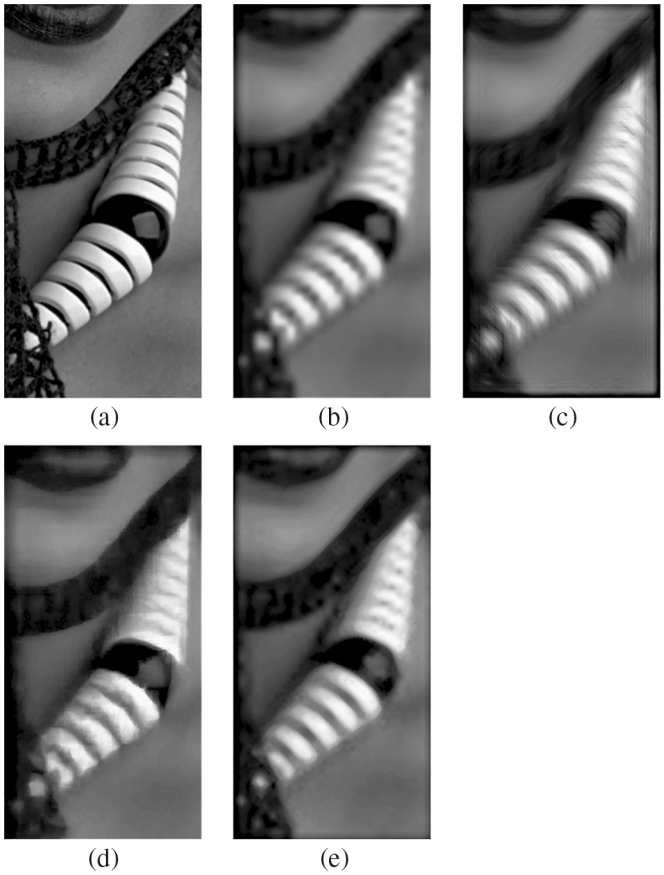

Fig. 10. Results of 8 × 8 expansion on a region of the "Face" image. (a) Original. (b) Cubic. (c) AQua. (d) Altamira. (e) MAP.

AQua in preserving edges, but not to the same extent as our algorithm, for instance in Fig. 9. This 8 × 8 expansion example shows that while fine textures and characters have little hope of being resolved in the expanded image (by any technique), it is possible to preserve the integrity of strong edges even with such a large expansion factor. The sharpness of some dominant edges in the MAP output is almost as high as the original! The sharpness of the biker's collar is preserved in our output, but lost in other methods. In all images, Altamira does a reasonable job of edge-preservation, but sometimes introduces unacceptable streaking and "leaking" artifacts causing objects to appear to "leak" across boundaries. Our output does not suffer from this problem. These results also seem to support the observation that MAP performs better edge-preservation in edges that registered during the prestep, but Altamira does better for edges that the prestep missed. This is entirely expected since MAP will smooth out regions it considers edgeless. MAP also exhibits some spurious high-frequency artifacts, a common feature of edge-preserving algorithms, and imposes a practical trade off between edge-preservation and clutter artifacts.

Our final results are of 4 × 4 expansion on a color image "Flowers." We show a zoomed-in portion of it in Fig. 12, comparing our result with Altamira, cubic and nearest-neighbor. AQuA result was marginally worse than cubic and has been omitted. The three colour channels were processed independently in cubic and MAP algorithms, and jointly by Altamira which takes colour images as direct input. Notice the presence of

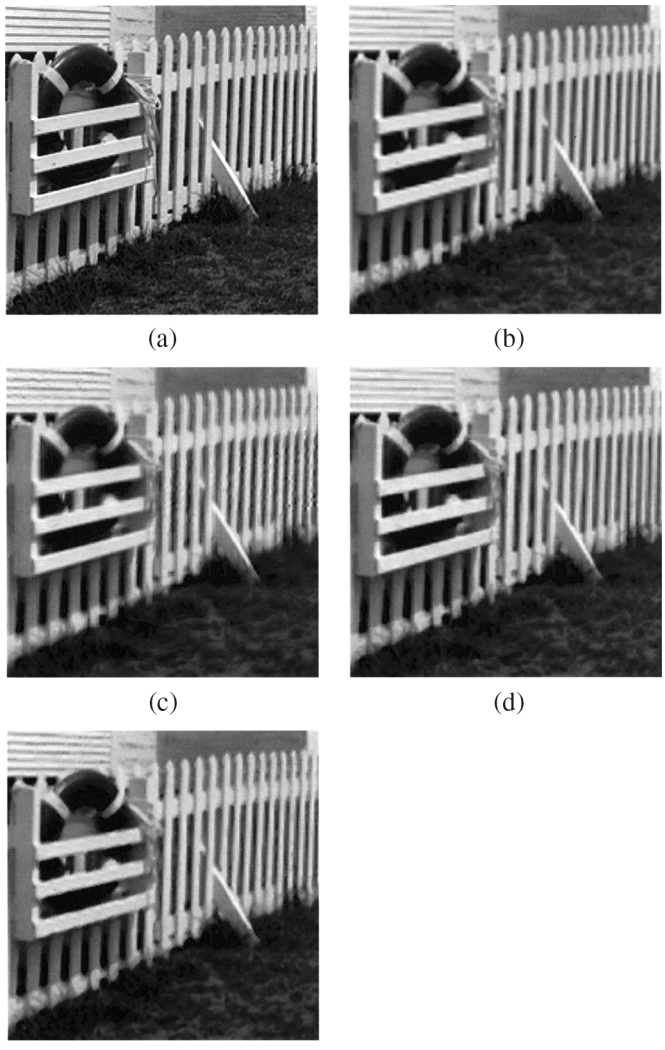

Fig. 11. Results of 2 × 2 expansion on a region of the "Lighthouse" image. (a) Original. (b) Cubic. (c) AQua. (d) Altamira. (e) MAP.

severe edge and color artifacts in the Altamira output, and their absence in MAP. Some uncontrolled leaking of strong edges appears in Altamira. Again, edge-preservation of MAP is superior at edges which were registered, and slightly worse in edges that were not. Compare for instance the vertical edge on the right of the image—it has a weak gradient and was not registered as an edge in our method. Altamira on the other hand is still able to resolve this edge.

The computation time of a MATLAB implementation of Algorithm A1 on 256 × 256 images was obtained. It took around 6 s for the linear-time solution $\mathbf{x}_2$ and 3 min for the full MAP solution $\mathbf{x}_{\mathrm{MAP}}$ on a 3.2-GHz Intel Xeon processor. Compared to this, a direct inversion of (7) via SVD on the same data took 27 min. MAP running time is likely to be reduced by many times in an optimized implementation. The theoretical flops counts supplied in Fig. 6 are probably much more useful in assessing the true numerical cost of our algorithm.

Numerical PSNR results are shown in Table I. It may be observed that PSNR of our results are higher than the other algorithms in most cases. AltaMira exhibits comparable performance in some cases, but does much worse than MAP-subspace

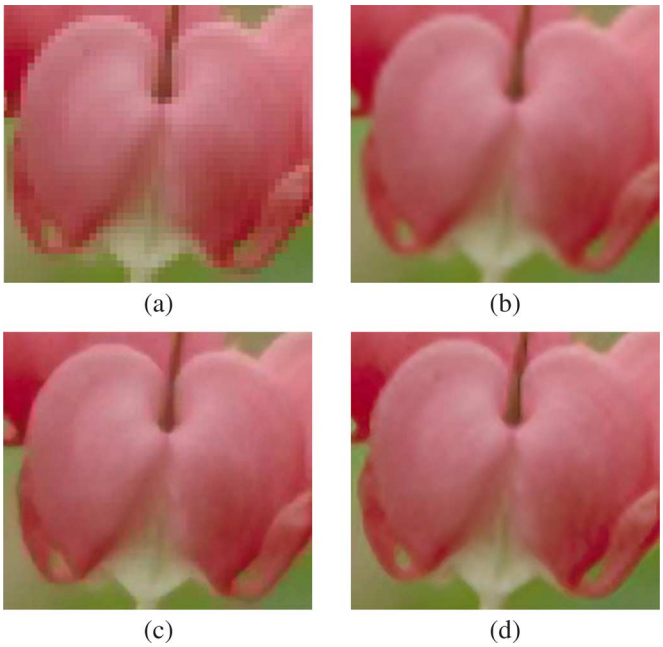

Fig. 12. Results of 4 × 4 expansion on a region of the "Flowers" image. (a) Original. (b) Cubic. (c) Altamira. (d) MAP.

TABLE I
PSNR PERFORMANCE OF EXPANSION ALGORITHMS. HIGHEST VALUES SHOWN IN BOLD

Image	Cubic	AQua	AltaMira	MAP-subspace
Bike	25.2	25.0	25.4	**26.4**
Face	32.1	31.0	32.0	**33.1**
Lighthouse	27.1	26.3	26.6	**27.8**
Flowers	37.6	37.3	37.4	**37.8**
Leaves	26.3	27.1	26.7	**27.2**
Lena	33.4	33.0	32.9	**34.4**
Rings (4x)	13.7	14.6	15.0	**17.0**
Bike (4x)	20.4	20.9	**21.2**	21.1

in some other cases. Our algorithm is the only one which consistently produces above-par results in almost all cases. We caution that a numerical metric may not truly reflect the perceptual quality of the results. The PSNR results are probably no more than a rough indicator of performance. Further, these results should be interpreted with due regard to the fact that all algorithms except MAP are "blind" to the decimation process (1).

The "Face" image has areas of mostly low-frequencies, and texture. The displayed region ("earrings") has several nonstraight edges, which shows up expansion artifacts like jagged edges, excessive smoothing and the like. The "Bike" image is slightly undersampled, with a lot of high frequencies and visible aliasing, thus making it a challenging case for any expansion algorithm. The "Lighthouse" image is quite (un)popular for reconstruction problems, because it tends to produce severe artifacts for almost all algorithms due to its numerous straight edges. We note that with the exception of "Lighthouse," our algorithm clearly outperforms the others in visual as well as metric terms (Table I), including the industry benchmark AltaMira.

V. DISCUSSION AND CONCLUSION

Since our method is geared towards image expansion rather than resampling, it is not well suited for fractional expansion factors or for applications involving image rotation or warping. Another limitation is in our edge-driven prior model, which leads to a diagonal weight matrix. However, there are other interesting models which may not have this property, and are likely to prove computationally challenging if used within our framework. While the probabilistic edge-driven prior model reduces the effect of incorrect hard decisions on edges, our results indicate that actual preservation of edges is usually possible only for large, dominant edges. Some recent work on image deblurring using edge-preserving MRF priors [36] has been reported, and it might be interesting to adapt it for expansion problems. The MRF approach incorporates piecewise smoothness assumptions and produces deblurred images without significant edge blurring. This is a consequence of the use of edge-preserving nonconvex MRF potential terms. However, due to nonconvexity the problem is extremely hard to solve and the use of approximate graph theoretic algorithms is necessary. The current approach is not able to handle MRF priors without losing computational tractability. However, our experimental evidence demonstrates clearly that even with these limitations our approach shows considerable promise compared to other popular methods.

Although the imaging model and the fast algorithm proposed in this paper are applicable only to dyadic expansion factors, we note that this is purely an implementation restriction rather than a mathematical one. It is possible to obtain results similar to the ones contained in this paper for any given integer expansion factor. However, a reading of Section III makes it apparent that for each expansion factor, a different fast algorithm is required. While powers of two expansion is certainly a restriction, it is probably the most frequently used, and several earlier works have focused on this case [8], [11]. Further work on obtaining a unified algorithm applicable for any expansion factor is ongoing.

In conclusion, this paper describes a new Bayesian approach to image expansion using an edge-driven prior image model. The resulting MAP estimation problem is derived. A new efficient algorithm based on subspace separation is proposed to make this problem computationally tractable. Separable 1-D processing and tiling are used to further reduce complexity. Results on real consumer images show visually noticeable and quantitatively measurable performance gains compared to other popular methods.

APPENDIX

A. Proof of Lemma 1

Using $P^H P = I$ in $H = P_e H_{\text{full}} = P_e D_N \Sigma D_N^H$, we get $H = P_e D_N \Sigma \left(\binom{P_e}{P_o} D_N\right)^H P$. Now we note some properties of

the DFT matrix. For $n, k = 0 \ldots N/2 - 1$

$$\left\{e^{-j\frac{2\pi}{N}2nk}\right\}_{n,k} = \left\{e^{-j\frac{2\pi}{N}2n(N/2+k)}\right\}_{n,k}$$
$$\left\{e^{-j\frac{2\pi}{N}(2n+1)k}\right\}_{n,k} = \left\{e^{-j\frac{2\pi}{N}k}\cdot e^{-j\frac{2\pi}{N}2nk}\right\}_{n,k}$$
$$\left\{e^{-j\frac{2\pi}{N}(2n+1)(N/2+k)}\right\}_{n,k} = -\left\{e^{-j\frac{2\pi}{N}k}\cdot e^{-j\frac{2\pi}{N}2nk}\right\}_{n,k}.$$

Using these results we obtain the partition $\begin{pmatrix} P_e \\ P_o \end{pmatrix} D_N = \begin{pmatrix} D_{N/2} & D_{N/2} \\ D_{N/2}E & -D_{N/2}E \end{pmatrix}$, where $D_{N/2}$ is the $N/2$-point DFT matrix, and $E = \mathrm{diag}(\{e^{-j(2\pi/N)k}\}_{k=0\ldots N-1})$. Then

$$H = (D_{N/2} \quad D_{N/2}) \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} \begin{pmatrix} D_{N/2} & D_{N/2} \\ D_{N/2}E & -D_{N/2}E \end{pmatrix}^H P$$

where Σ_1 and Σ_2 correspond to the two halves of the DFT of h_{full}. To get the SVD of H equate above with Lemma 1 and assign $\Sigma_H V_1^H = ((\Sigma_1 + \Sigma_2) D_{N/2}^H E^*(\Sigma_1 - \Sigma_2) D_{N/2}^H) P$. The rest of proof follows by using $V_1^H V_1 = I_{N/2}$. To obtain V_2, it is straightforward to select a unitary matrix orthogonal to V_1

$$V_2 = P^T \begin{pmatrix} D_{N/2}\Sigma_H^{-1}(\Sigma_1 - \Sigma_2) \\ -D_{N/2}E\Sigma_H^{-1}(\Sigma_1 + \Sigma_2) \end{pmatrix}. \tag{17}$$

B. Theorem 1: Approximate Diagonalization Resulting From Sparse Prior

In the system $(\Gamma + C)\binom{\eta_1}{\eta_2} = \underline{\mathrm{y}}$, C_{ij} are such that $\|C_{ij}\|_F < \sqrt{\kappa}\|\Gamma_{ij}\|_F$, with diagonal Γ_{ij}

$$\Gamma_{11} = \Sigma_H^2 + 4\mu^2\Sigma_H^{-2}\left(\Sigma_{d1}^2\Sigma_1^2 + \Sigma_{d2}^2\Sigma_2^2\right)$$
$$\Gamma_{22} = 4\mu^2\Sigma_H^{-2}\left(\Sigma_{d1}^2\Sigma_2^2 + \Sigma_{d2}^2\Sigma_1^2\right)$$
$$\Gamma_{12} = \Gamma_{21} = -4\mu^2\Sigma_H^{-2}\Sigma_1\Sigma_2\left(\Sigma_{d1}^2 - \Sigma_{d2}^2\right).$$

Proof: We will derive expressions for Γ_{22} and C_{22}, and show that $\|C_{22}\|_F^2 < \kappa\|\Gamma_{22}\|_F^2$. Other results follow similarly. Now, the SVD of $\mathcal{D}$ is $\mathcal{D} = D_N\Sigma_d D_N^H$; hence, from (17)

$$GV_2 = \mu W^{-1}\mathcal{D}V_2 = W^{-1}D_N\begin{bmatrix}\Lambda_2^1 \\ \Lambda_2^2\end{bmatrix}$$

where $\Lambda_2^1 = 2\mu\Sigma_H^{-1}\Sigma_{d1}\Sigma_1$; $\Lambda_2^2 = 2\mu\Sigma_H^{-1}\Sigma_{d2}\Sigma_2$. Then

$$V_2^H Q V_2 = \begin{bmatrix}\Lambda_2^1 & \Lambda_2^2\end{bmatrix} D_N^H W^{-2} D_N \begin{bmatrix}\Lambda_2^1 \\ \Lambda_2^2\end{bmatrix}.$$

Let $W^{-2} = I - \overline{W}^2$ be a mostly zero diagonal matrix with $0 \le \overline{W}_{i,i} \le 1$, $i \in [0, N-1]$. Then

$$V_2^H Q V_2 = \Gamma_{22} - \begin{bmatrix}\Lambda_2^1 & \Lambda_2^2\end{bmatrix} D_N^H \overline{W}^2 D_N \begin{bmatrix}\Lambda_2^1 \\ \Lambda_2^2\end{bmatrix}$$

with $\Gamma_{22} = [\Lambda_2^1 \ \Lambda_2^2]\left[\begin{smallmatrix}\Lambda_2^1 \\ \Lambda_2^2\end{smallmatrix}\right] = 4\mu^2\Sigma_H^{-2}(\Sigma_{d1}^2\Sigma_2^2 + \Sigma_{d2}^2\Sigma_1^2)$. Name the right term above C_{22}. Then

$$\|C_{22}\|_F \le \left\|\begin{bmatrix}\Lambda_2^1 & \Lambda_2^2\end{bmatrix}\right\|_F^2 \|D_N\|_F^2 \|\overline{W}^2\|_F$$

by matrix norm property. It is easy to verify that $\left\|\begin{bmatrix}\Lambda_2^1 & \Lambda_2^2\end{bmatrix}\right\|_F^2 = \|\Gamma_{22}\|_F$, and $\|D_N\|_F = 1$. Finally, $\|\overline{W}^2\|_F^2 = tr(\overline{W}^2)/N < \kappa$. Therefore

$$\|C_{22}\|_F^2 < \kappa\|\Gamma_{22}\|_F^2.$$

Proceeding similarly with $V_1^H Q V_1$, $V_1^H Q V_2$, etc., the other norm results follow. Further, the right hand side of (12) becomes

$$-\begin{pmatrix} V_1^H Q x_0 \\ V_2^H Q x_0 \end{pmatrix} = -\begin{pmatrix} V_1^H Q V_1 \\ V_2^H Q V_2 \end{pmatrix} \Sigma_H^{-1} D_{N/2}^H \mathrm{y} = \underline{\mathrm{y}}.$$

■

C. Interpretation of x_3 as a Constrained MAP Solution

Let Z span the nullspace of H, i.e., $HZ = 0$, and let x_0 satisfy $H\mathrm{x}_0 = \mathrm{y}$. Then, it is easily shown from the SVD results of Lemma 1 that

$$\mathrm{x}_3 = \mathrm{x}_0 - Z(Z^T G^T G Z)^{-1} Z^T G^T G \mathrm{x}_0.$$

This is exactly the well-known constrained least squares solution (see [32] for reference)

$$\mathrm{x}_3 = \arg\min \|G\mathrm{x}\|^2, \quad \text{given } H\mathrm{x}_0 = \mathrm{y}.$$

REFERENCES

[1] E. Meijering, "A chronology of interpolation," *Proc. IEEE*, vol. 90, no. 3, pp. 319–342, Mar., 2002.

[2] S. Andrews and F. Harris, "Polynomial approximations of interpolants," in *Proc. 33rd Asilomar Conf. Signals, Systems, and Computers*, 1999, vol. 1, pp. 447–451.

[3] T. Blu, P. Thvenaz, and M. Unser, "Interpolation revisited," *IEEE Trans. Med. Imag.*, vol. 19, no. 7, pp. 739–758, Jul. 2000.

[4] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, no. 6, pp. 1153–1160, Dec. 1981.

[5] L. Condat, T. Blu, and M. Unser, "Beyond interpolation: Optimal reconstruction by quasi-interpolation," in *Proc. IEEE Int. Conf. Image Processing*, 2005, pp. 33–36.

[6] S. Lee and J. Paik, "Image interpolation using adaptive fast b-spline filtering," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 1993, pp. 177–180.

[7] K. Ratakonda and N. Ahuja, "POCS-based adaptive image magnification," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 1998, vol. 3, pp. 203–207.

[8] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," *IEEE Trans. Image Process.*, vol. 4, no. 3, pp. 285–295, Mar. 1995.

[9] J. Allebach and P. Wong, "Edge-directed interpolation," in *Proc. IEEE Int. Conf. Image Processing*, 1996, pp. 707–710.

[10] G. Ramponi, "Warped distance for space-variant linear image interpolation," *IEEE Trans. Image Process.*, vol. 8, no. 5, pp. 629–639, May 1999.

[11] X. Li and M. Orchard, "New edge directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.

[12] S. Citrin and M. R. Azimi-Sadjadi, "A full-plane block kalman filter for image restoration," *IEEE Trans. Image Process.*, vol. 1, no. 4, pp. 488–495, Apr. 1992.

[13] K. Erler and E. Jernigan, "Adaptive recursive image filtering," *Proc. SPIE*, pp. 3017–3021, 1991.

[14] D. Muresan and T. Parks, "Adaptive optimal recovery image interpolation," *IEEE Trans. Image Process.*, vol. 13, no. 5, pp. 690–698, May 2004.

[15] R. Schultz and R. Stevenson, "A Bayesian approach to image expansion for improved definition," *IEEE Trans. Image Process.*, vol. 3, no. 3, pp. 233–242, May 1994.

[16] R. Chan, T. Chan, L. Shen, and Z. Shen, "Wavelet algorithms for high-resolution image reconstruction," *SIAM J. Sci. Comput.*, vol. 24, no. 4, pp. 1408–1432, 2003.
[17] R. Molina, M. Vega, J. Abad, and A. Katsaggelos, "Parameter estimation in bayesian high-resolution image reconstruction with multisensors," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1655–1667, Dec. 2003.
[18] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1646–1658, Dec. 1997.
[19] N. Bose and K. Boo, "High-resolution image reconstruction with multisensors," *Int. J. Imag. Sci. Technol.*, vol. 9, pp. 141–163, 1998.
[20] H. A. Aly and E. Dubois, "Specification of the observation model for regularized image up-sampling," *IEEE Trans. Image Process.*, vol. 14, no. 5, pp. 567–576, May 2005.
[21] T. Berber, J. Stromberg, and T. Eltoft, "Adaptive regularized constrained least squares image restoration," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1191–1203, Sep. 1999.
[22] A. Katsaggelos, J. Biermond, R. Schafer, and R. Mersereau, "A regularized iterative image restoration algorithm," *IEEE Trans. Signal Process.*, vol. 39, no. 4, pp. 914–929, Apr. 1991.
[23] S. Li, *Markov Random Field Modeling in Computer Vision*. New York: Springer-Verlag, 1995.
[24] V. Z. Mesarovic, N. P. Galatsanos, and M. N. Wernick, "Iterative maximum *a posteriori* (MAP) restoration from partially-known blur for tomographic reconstruction," in *Proc. IEEE Int. Conf. Image Processing*, 1995, pp. 512–515.
[25] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
[26] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 6, pp. 849–865, Nov. 1988.
[27] C. Bouman and K. Sauer, "A generalized Gaussian image model for edge preserving MAP estimation," *IEEE Trans. Image Process.*, vol. 2, no. 3, pp. 296–310, Jul. 1993.
[28] D. W. Paglieroni, "Distance transforms: Properties and machine vision applications," *Comput. Vis., Graph., Image Process.: Graph. Models Image Process.*, vol. 54, no. 1, pp. 56–74, 1992.
[29] A. Thompson, J. Brown, J. Kay, and M. Titterington, "A study of methods for choosing the smoothing parameter in image restoration by regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 5, pp. 326–339, May 1991.
[30] M. Chen and T. Pavlidis, "Image seaming for segmentation on parallel architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 6, pp. 588–594, Jun. 1990.
[31] R. M. Gray, *Toeplitz and Circulant Matrices: A Review*. Boston, MA: Now Publishers, 2006.
[32] G. Golub and C. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1996.
[33] Altamira, Commercial, Fractal Based Interpolation Algorithm [Online]. Available: http://www.genuinefractals.com
[34] D. Muresan and T. Parks, "Adaptive optimal recovery image interpolation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 2001, pp. 1949–1952.
[35] W. Freeman, T. Jones, and E. Pasztor, "Example-based super-resolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, Mar./Apr. 2002.
[36] A. Raj and R. Zabih, "A graph cut algorithm for generalized image deconvolution," in *Proc. Int. Conf. Computer Vision*, 2005, pp. 1901–1909.

Ashish Raj was born in Godda, India. He received the B.S. degree (with honors) in electrical engineering from the University of Auckland, New Zealand, in November 1997, and the Ph.D. degree in electrical and computer engineering from Cornell University, Ithaca, NY, in 2004.

Since 2005, he has been working on medical imaging at the Radiology Department, University of California at San Francisco, where he is an Assistant Professor. His research interests are centered around inverse problems that arise in signal processing, medical imaging, and telecommunications.

Kailash Thakur received the Ph.D. and the D.Sc. degrees in physics from the University of Allahabad, India, in 1974 and 1988, respectively.

From 1978 to 1980, he was a Postdoctoral Fellow in solid state physics at the University of Warwick, U.K. He was a Professor at Bhagalpur University, India, and Asmara University, Eritrea. Since 1993, he has been a Senior Scientist with the New Zealand Institute for Industrial Research, where he has worked on various research projects in microwave and industrial vision.