Hindawi Publishing Corporation
EURASIP Journal on Applied Signal Processing
Volume 2006, Article ID 61859, Pages 1–15
DOI 10.1155/ASP/2006/61859

Video-to-Video Dynamic Super-Resolution for Grayscale and Color Sequences

Sina Farsiu,[1] Michael Elad,[2] and Peyman Milanfar[1]

[1] *Electrical Engineering Department, University of California Santa Cruz, Santa Cruz, CA 95064, USA*
[2] *Computer Science Department, Technion – Israel Institute of Technology, Haifa 32000, Israel*

Received 17 December 2004; Revised 10 March 2005; Accepted 15 March 2005

We address the dynamic super-resolution (SR) problem of reconstructing a high-quality set of monochromatic or color super-resolved images from low-quality monochromatic, color, or mosaiced frames. Our approach includes a joint method for simultaneous SR, deblurring, and demosaicing, this way taking into account practical color measurements encountered in video sequences. For the case of translational motion and common space-invariant blur, the proposed method is based on a very fast and memory efficient approximation of the Kalman filter (KF). Experimental results on both simulated and real data are supplied, demonstrating the presented algorithms, and their strength.

1. INTRODUCTION

Theoretical and practical limitations usually constrain the achievable resolution of any imaging device. While higher-quality images may result from more expensive imaging systems, often we wish to increase the resolution of images previously captured under nonideal situations. For instance, enhancing the quality of a video sequence captured by surveillance cameras in a crime scene is an example of these situations.

The basic idea behind SR is the fusion of a sequence of low-resolution (LR) noisy blurred images to produce a higher-resolution image. Early works on SR showed that it is the aliasing effects in the LR images that enable the recovery of the high-resolution (HR) fused image, provided that a relative subpixel motion exists between the undersampled input images [1]. However, in contrast to the clean but practically naive frequency-domain description of SR in that early work, in general, SR is a computationally complex and numerically ill-posed problem in many instances [2]. In recent years, more sophisticated SR methods have been developed (see [2–12] as representative works).

In this work, we consider SR applied on an image sequence, producing a sequence of SR images. At time point t, we desire an SR result that fuses the causal images at times $t, t-1, \ldots, 1$. The natural approach, as most existing works so far suggest, is to apply the regular SR on this set of images with the tth frame as a reference, produce the SR output, and repeat this process all over again per each temporal point. We refer to this as the static SR method, since it does not exploit the temporal evolution of the process.

In contrast, in this work, we adopt a dynamic point of view, as introduced in [13, 14], in developing the new SR solution. The memory and computational requirements for the static process are so taxing as to preclude its direct application to the dynamic case, without highly efficient algorithms. It is natural to expect that if the SR problem is solved for time $t-1$, our task for time t could use the solution at the previous time instant as a stepping stone towards a faster and more reliable SR solution. This is the essence of how dynamic SR is to gain its speed and better results, as compared to a sequence of detached static SR solutions.

The work presented here builds on the core ideas as appeared in [13, 14], but deviates from them in several important ways, to propose a new and better reconstruction algorithm.

(i) *Speed*. Whereas the methods in [13, 14] rely on the information pair to approximate the KF, this work uses the more classic mean-covariance approach. We show that for the case of translational motion and common space-invariant blur, the proposed method is computationally less complex than the dynamic SR methods proposed previously. Also, in line with [15], we show that this problem can be decomposed into two disjoint pieces, without sacrificing optimality.

(ii) *Treatment of mosaiced images*. In this paper, we focus on two common resolution-enhancement problems in

digital video/photography that are typically addressed separately, namely, SR and demosaicing. While SR is naturally described for monochrome images, aiming to increase resolution by the fusion of several frames, demosaicing is meant to recover missing color values, decimated deliberately by the sensor. In this work, we propose a method of dealing with these two problems jointly, and dynamically. Note that in our previous work as appeared in [16, 17] we addressed the static multiframe demosaicing problem, and so the work presented here stands as an extension of it to the dynamic case.

(iii) *Treatment of color*. Our goal in this paper is to develop a dynamic SR algorithm for both monochromatic and color input and output sequences. We seek improvements in both visual quality (resolution enhancement and color artifact reduction) and computational/memory efficiency. We introduce advanced priors that handle both spatial and color-wise relationships properly, this way leading to high quality recovery.

(iv) *Causality*. The work presented in [13, 14] considered a causal mode of operation, where the output image at time t_0 fuses the information from times $t \le t_0$. This is the appropriate mode of operation when online processing is considered. Here, we also study a noncausal processing mode, where every HR reconstructed image is derived as an optimal estimate incorporating information from *all* the frames in the sequence. This is an appropriate mode of operation for offline processing of movies, stored on disk. We use the smoothed KF to obtain an efficient algorithm for this case.

This paper is organized as follows. In Section 2, we discuss a fast dynamic image fusion method for the translational motion model, assuming regular monochromatic images, considering both causal and noncausal modes. This method is then extended in Section 3 to consider an enhancement algorithm of monochromatic deblurring and interpolation. We address multiframe demosaicing and color-SR deblurring problems in Section 4. Simulations on both real and synthetic data sequences are presented in Section 5, and Section 6 concludes this paper.

Before delving into the details, we should mention that this paper (with all color pictures and a Matlab-based software package for resolution enhancement) is available at http://www.soe.ucsc.edu/~milanfar.

2. DYNAMIC DATA FUSION

2.1. Recursive model

In this paper, we use a general linear dynamic forward model for the SR problem as in [13, 14]. A dynamic scene with intensity distribution $\underline{X}(t)$ is seen to be warped at the camera lens because of the relative motion between the scene and camera, and blurred by camera lens and sensor integration. Then, it is discretized at the CCD, resulting in a digitized noisy frame $\underline{Y}(t)$. Discretization in many commercial digital cameras is a combination of color filtering and downsampling processes. However, in this section, we will restrict our treatment to simple monochrome imaging. We represent this forward model by the following state-space equations [18]:

$$\underline{X}(t) = F(t)\underline{X}(t-1) + \underline{U}(t), \tag{1}$$

$$\underline{Y}(t) = D(t)H(t)\underline{X}(t) + \underline{W}(t). \tag{2}$$

Equation (1) describes how the ideal superresolved images relate to each other through time. We use the underscore notation such as $\underline{X}$ to indicate a vector derived from the corresponding image of size $[rQ_1 \times rQ_2]$ pixels, scanned in lexicographic order. The current image $\underline{X}(t)$ is of size $[r^2Q_1Q_2 \times 1]$, where r is the resolution-enhancement factor, and $[Q_1 \times Q_2]$ is the size of an input LR image. Equation (1) states that up to some innovation content $\underline{U}(t)$, the current HR image is a geometrically warped version of the previous image, $\underline{X}(t-1)$. The $[r^2Q_1Q_2 \times r^2Q_1Q_2]$ matrix $F(t)$ represents this warp operator. The so-called system noise $\underline{U}(t)$, of size $[r^2Q_1Q_2 \times 1]$, is assumed to be additive zero-mean Gaussian with $C_u(t)$ as its covariance matrix of size $[r^2Q_1Q_2 \times r^2Q_1Q_2]$. Note that the closer the overlapping regions of $\underline{X}(t)$ and the motion compensated $\underline{X}(t-1)$ are, the smaller $C_u(t)$ becomes. Therefore, $C_u(t)$ reflects the accuracy of the motion estimation process and for overlapped regions it is directly related to the motion estimation covariance matrix.

As to equation (2), it describes how the measured image $\underline{Y}(t)$ of size $[Q_1Q_2 \times 1]$ is related to the ideal one, $\underline{X}(t)$. The camera's point spread function (PSF) is modelled by the $[r^2Q_1Q_2 \times r^2Q_1Q_2]$ blur matrix $H(t)$, while the $[Q_1Q_2 \times r^2Q_1Q_2]$ matrix $D(t)$ represents the downsampling operation at the CCD (downsampling by the factor r in each axis). In mosaiced cameras, this matrix also represents the effects of the color filter array, which further downsamples the color images—this will be described and handled in Section 4. The noise vector $\underline{W}(t)$ of size $[Q_1Q_2 \times 1]$ is assumed to be additive, zero-mean, white Gaussian noise. Thus, its $[Q_1Q_2 \times Q_1Q_2]$ covariance matrix is $C_w(t) = \sigma_w^2 I$. We further assume that $\underline{U}(t)$ and $\underline{W}(t)$ are independent of each other.

The equations given above describe a system in its *state-space* form, where the state is the desired ideal image. Thus, a KF formulation can be employed to recursively compute the optimal estimates $(\underline{X}(t),\ t \in \{1, \ldots, N\})$ from the measurements $(\underline{Y}(t),\ t \in \{1, \ldots, N\})$, assuming that $D(t), H(t)$, $F(t), \sigma_w$, and $C_u(t)$ are all known [13, 14, 18]. This estimate could be done causally, as an online processing of an incoming sequence, or noncausally, assuming that the entire image sequence is stored on disk and processed offline. We consider both these options in this paper.

As to the assumption about the knowledge of various components of our model, while each of the operators $D(t), H(t)$, and $F(t)$ may vary in time, for most situations the downsampling (and later color filtering), and camera blurring operations remain constant over time assuming that the images are obtained from the same camera. In this paper, we further assume that the camera PSF is space-invariant, and the motion is composed of pure translations, accounting for

either vibrations of a gazing camera, or a panning motion of a faraway scene. Thus, both H and $F(t)$ are block-circulant matrices,[1] and as such, they commute. We assume that H is known, being dependent on the camera used, and $F(t)$ is built from motion estimation applied on the raw sequence $\underline{Y}(t)$. The downsampling operator D is completely dictated by the choice of the resolution-enhancement factor (r). As to σ_w, and $C_u(t)$, those will be handled shortly.

We limit our model to the case of translational motion for several reasons. First, as we describe later, such a motion model allows for an extremely fast and memory efficient dynamic SR algorithm. Second, while simple, the model fairly well approximates the motion contained in many image sequences, where the scene is stationary and only the camera moves in approximately linear fashion. Third, for sufficiently high frame rates, most motion models can be (at least locally) approximated by the translational model. Finally, we believe that an in-depth study of this simple case yields much insight into the more general cases of motion in dynamic SR.

By substituting $\underline{Z}(t) = H\underline{X}(t)$, we obtain from (1) and (2) an alternative model, where the state vector is $\underline{Z}(t)$,

$$\underline{Z}(t) = F(t)\underline{Z}(t-1) + \underline{V}(t), \tag{3}$$

$$\underline{Y}(t) = D\underline{Z}(t) + \underline{W}(t). \tag{4}$$

Note that the first of the two equations is obtained by left multiplication of both sides of (1) by H and using the fact that it commutes with $F(t)$. Thus, the vector $\underline{V}(t)$ is a colored version of $\underline{U}(t)$, leading to $C_v(t) = HC_u(t)H^T$ as the covariance matrix.

With this alternative definition of the state of the dynamic system, the solution of the inverse problem at hand decomposes, without loss of optimality, into the much simpler subtasks of fusing the available images to compute the estimated blurry image $\hat{\underline{Z}}(t)$, followed by a deblurring/interpolation step, estimating $\hat{\underline{X}}(t)$ from $\hat{\underline{Z}}(t)$. In this section, we treat the three color bands separately. For instance, only the red band values in the input frames, $\underline{Y}(t)$, contribute to the reconstruction of the red band values in $\hat{\underline{Z}}(t)$. The correlation of the different color bands is discussed and exploited in Section 4.

We next study the application of KF to estimate $\underline{Z}(t)$. In general, the application of KF requires the update of the state vector's covariance matrix per each temporal point, and this update requires an inversion of the state vector's covariance matrix. For a superresolved image with $r^2Q_1Q_2$ pixels, this matrix is of size $[r^2Q_1Q_2 \times r^2Q_1Q_2]$, implying a prohibitive amount of computations and memory.

Fast and memory efficient alternative ways are to be found, and such methods were first proposed in the context of the dynamic SR in [13, 14]. Here we show that significant further speedups are achieved for the case of translational motion and common space-invariant blur.

[1] True for cyclic boundary conditions that will be assumed throughout this work.

2.2. Forward data fusion method

The following defines the forward Kalman propagation and update equations [18] that accounts for a causal (online) process. We assume that at time $t-1$ we already have the mean-covariance pair, $(\hat{\underline{Z}}(t-1), \widehat{M}(t-1))$, and those should be updated to account for the information obtained at time t. We start with the covariance matrix update based on (3),

$$\widetilde{M}(t) = F(t)\widehat{M}(t-1)F^T(t) + C_v(t). \tag{5}$$

The KF gain matrix is given by

$$K(t) = \widetilde{M}(t)D^T\left[C_w(t) + D\widetilde{M}(t)D^T\right]^{-1}. \tag{6}$$

This matrix is rectangular of size $[r^2Q_1Q_2 \times Q_1Q_2]$. Based on $K(t)$, the updated state-vector mean is computed by

$$\hat{\underline{Z}}(t) = F(t)\hat{\underline{Z}}(t-1) + K(t)\left[\underline{Y}(t) - DF(t)\hat{\underline{Z}}(t-1)\right]. \tag{7}$$

The final stage requires the update of the covariance matrix, based on (4),

$$\widehat{M}(t) = \mathrm{Cov}\left(\hat{\underline{Z}}(t)\right) = \left[\mathbf{I} - K(t)D\right]\widetilde{M}(t). \tag{8}$$

More on the meaning of these equations and how they are derived can be found in [18, 19].

While in general the above equations require the propagation of intolerably large matrices in time, if we refer to $C_v(t)$ as a diagonal matrix, then $\widetilde{M}(t)$ and $\widehat{M}(t)$ are diagonal matrices of size $[r^2Q_1Q_2 \times r^2Q_1Q_2]$. It is relatively easy to verify this property: for an arbitrary diagonal matrix G_B (B stands for *big*), the matrix DG_BD^T is a diagonal matrix. Similarly, for an arbitrary diagonal matrix G_S (S stands for *small*), the matrix D^TG_SD is diagonal as well. Also, in [15], it is shown that for an arbitrary pure translation matrix F and an arbitrary diagonal matrix G_B, the matrix FG_BF^T is diagonal. Therefore, if the matrix $\widetilde{M}(0)$ is initialized as a diagonal matrix, then $\widetilde{M}(t)$ and $\widehat{M}(t)$ are necessarily diagonal for all t, being the results of summation, multiplication, and inversions of diagonal matrices.

Diagonality of $C_v(t)$ is a key assumption in transferring the general KF into a simple and fast procedure, and as we will see, the approximated version emerging is quite faithful. Following [13, 14], if we choose a matrix $\sigma_v^2\mathbf{I} \geq C_v(t)$, it implies that $\sigma_v^2\mathbf{I} - C_v(t)$ is a positive semidefinite matrix, and there is always a finite σ_v that satisfies this requirement. Replacing $C_v(t)$ with this majorizing diagonal matrix, the new state-space system in (3) and (4) simply assumes a stronger innovation process. The effect on the KF is to rely less on the temporal relation in (3) and more on the measurements in (4). In fact, at the extreme case, if $\sigma_v \to \infty$, the KF uses only the measurements, leading to an intraframe maximum-likelihood estimator. Thus, more generally, such a change causes a loss in the accuracy of the KF because it relies less on the internal dynamics of the system, but this comes with a welcomed simplification of the recursive estimator. It must be clear that such change in $C_v(t)$ has no impact on the convergence properties of the dynamic estimator we apply, and

it does not introduce a bias in the estimate. Note that all the above is true also for a diagonal non-Toeplitz alternative, where the main diagonal entries are varying in space.

Once we chose $C_v(t)$ to be diagonal, (5), (6), (7), and (8) are simplified, and their use is better understood on a pixel-by-pixel basis. Before we turn to describe such a KF for the forward case, we introduce some notations to simplify the explanation of the process.

The warp matrix $F(t)$ and its transpose can be exactly interpreted as image shift operators [8, 15]. We use hereafter the superscript "f," to simplify the notation of forward shifting of vectors and diagonal matrices, and thus $\underline{Z}^f(t) = F(t)\underline{Z}(t-1)$ and $\widehat{M}^f(t) = F(t)\widehat{M}(t-1)F^T(t)$.

Also, the matrix D and its transpose can be exactly interpreted as downsampling and upsampling operators. Application of $D\underline{Z}(t)$ and $D\widehat{M}(t)D^T$ results in downsampling of the vector $\underline{Z}(t)$ and the diagonal matrix $\widehat{M}(t)$. Likewise, application of $D^T\underline{Y}(t)$ and $D^TC_w(t)D$ results in upsampling of the vector $\underline{Y}(t)$ and the diagonal matrix $C_w(t)$ with zero filling. Figure 1 illustrates the effect of matrix upsampling and downsampling operations, and this also sheds some light on the previous discussion on the diagonality assumption on $\widetilde{M}(t)$ and $\widehat{M}(t)$.

Finally, we will use the notation $[G]_q$ to refer to the (q,q) entry of the diagonal matrix G, and $[\underline{G}]_q$ to refer to the $(q,1)$ entry in the vector $\underline{G}$. This way we will be able to handle both the LR and the HR grids in the same equations.

Let us now return to the KF equations and show how they are implemented in practice on a pixel-by-pixel basis. First, referring to the propagated covariance matrix, we start by observing that in (6), the term $C_w(t)+D\widetilde{M}D^T$ is a diagonal matrix of size $[Q_1Q_2 \times Q_1Q_2]$, with the (q,q)th entry being

$$[C_w(t)]_q + [\widehat{M}^f(t)]_{qr^2} + [C_v(t)]_{qr^2}, \tag{9}$$

with q in the range $[1, Q_1Q_2]$. The "jumps" in r^2 in the indices of $\widehat{M}^f(t)$ and $C_v(t)$ are caused by the decimation D. Applying an inversion replaces the above by its reciprocal. Using interpolation $D^T(C_w(t)+D\widetilde{M}D^T)^{-1}D$ gives a diagonal matrix of size $[r^2Q_1Q_2 \times r^2Q_1Q_2]$, with the qth entry being

$$\frac{1}{[C_w(t)]_{q/r^2} + [\widehat{M}^f(t)]_q + [C_v(t)]_q}, \tag{10}$$

this time referring to the indices $q = r^2, 2r^2, \ldots, Q_1Q_2r^2$. For all other $(r^2-1)Q_1Q_2$ indices, the entries are simply zeros, filled by the interpolation. Merging this with (6) and (8), we obtain

$$[\widehat{M}(t)]_q = \begin{cases} \dfrac{[C_w(t)]_{q/r^2}([\widehat{M}^f(t)]_q + [C_v(t)]_q)}{[C_w(t)]_{q/r^2} + [\widehat{M}^f(t)]_q + [C_v(t)]_q} \\ \qquad \text{for } q = r^2, 2r^2, \ldots, Q_1Q_2r^2, \\ [\widehat{M}^f(t)]_q + [C_v(t)]_q \quad \text{otherwise.} \end{cases} \tag{11}$$

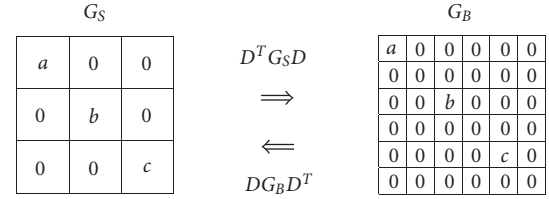


Figure 1: The diagonal matrix G_B on the right is the result of applying the upsampling operation (D^TG_SD) on an arbitrary diagonal matrix G_S on the left. The matrix G_S can be retrieved by applying the downsampling operation (DG_BD^T). The upsampling/downsampling factor for this example is two.

Note that the incorporation of each newly measured LR image only updates values of Q_1Q_2 entries in the diagonal of $\widehat{M}(t)$, located at the $[r^2, 2r^2, \ldots, r^2Q_1Q_2]$ positions. The remaining $(r^2-1)Q_1Q_2$ diagonal entries are simply propagated from the previous temporal point, based on (5) only. As we will see, the same effect holds true for the update of $\underline{\hat{Z}}(t)$, where $(r^2-1)Q_1Q_2$ entries are propagated from the previous temporal point without an update.

Turning to the update of the mean vector, $\underline{\hat{Z}}(t)$, using the same reasoning applied on (6) and (7), we obtain the relation

$$[\underline{\hat{Z}}(t)]_q = \begin{cases} \dfrac{[C_w(t)]_{q/r^2}[\underline{\hat{Z}}^f(t)]_q + ([\widehat{M}^f(t)]_q + [C_v(t)]_q)[\underline{Y}(t)]_{q/r^2}}{[C_w(t)]_{q/r^2} + [\widehat{M}^f(t)]_q + [C_v(t)]_q} \\ \qquad \text{for } q = r^2, 2r^2, \ldots, Q_1Q_2r^2, \\ [\underline{\hat{Z}}^f(t)]_q \quad \text{otherwise.} \end{cases} \tag{12}$$

Figure 2 describes the above equation's upper part as a block diagram. Notice that two images are merged here—an interpolated version of $\underline{Y}(t)$ and $\underline{\hat{Z}}^f(t)$. The merging is done as a weighted average between the two, as the figure suggests.

The overall procedure using these update equations is outlined in Algorithm 1. Since the update operations are simply based on shifting the previous estimates $\underline{\hat{Z}}(t-1)$ and $\widehat{M}(t-1)$ and updating the proper pixels using (11) and (12), we refer hereafter to this algorithm as the dynamic shift-and-add process. Similarly, we call $\underline{\hat{Z}}(t)$ the dynamic shift-and-add image. Several comments are in order, regarding the above procedure.

(1) Initialization. For long enough sequences, the initialization choice has a vanishing effect on the outcome. Choosing $\widehat{M}(0) = \epsilon^2\mathbf{I}$ guarantees that $\widehat{M}(t)$ is strictly positive definite at all times, where ϵ is an arbitrary large number ($\epsilon \gg \sigma_w^2$). Better initialization can be proposed, based on interpolation of the image $\underline{Y}(t)$. The same applies to regions coming from occlusion—those can be initialized by the current image.

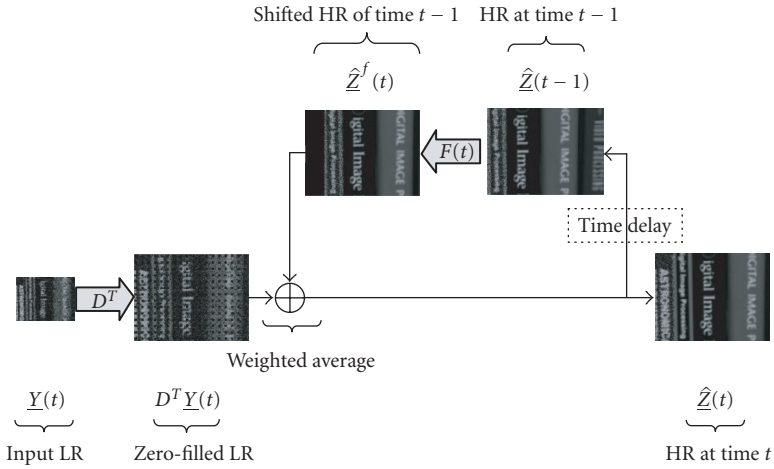


Figure 2: Block diagram representation of (12), where $\hat{\underline{Z}}(t)$, the new input HR output frame, is the weighted average of $\underline{Y}(t)$, the current input LR frame, and $\hat{\underline{Z}}^f(t)$, the previous estimate of the HR image after motion compensation.

(i) Task. Given $\{\underline{Y}(t)\}_{t\geq 1}$, estimate $\{\underline{Z}(t)\}_{t\geq 1}$ causally.
(ii) Initialization. Set $t = 0$, choose $\hat{\underline{Z}}(t) = \underline{0}$ and $\widehat{M}(t) = \epsilon^2\mathbf{I}$.
(iii) Update process. Set $t \to t+1$, obtain $\underline{Y}(t)$, and apply

(1) motion compensation: compute $\hat{\underline{Z}}^f(t) = F(t)\hat{\underline{Z}}(t-1)$ and $\widehat{M}^f(t) = F(t)\widehat{M}(t-1)F^T(t)$;
(2) update of the covariance: use (11) to compute the update $\widehat{M}(t)$;
(3) update of the mean: use (12) to compute the update $\hat{\underline{Z}}(t)$.

(iv) Repeat. Update process.

Algorithm 1: Forward dynamic shift-and-add algorithm.

(2) Arrays propagated in time. The algorithm propagates two images in time, namely, the image estimate $\hat{\underline{Z}}(t)$, and the main diagonal of its covariance matrix $\widehat{M}(t)$. This last quantity represents the weights assigned per pixel for the temporal fusion process, where the weights are derived from the accumulated measurements for the pixel in question.

At this point, we have an efficient recursive estimation algorithm producing estimates of the blurry HR image sequence $\hat{\underline{Z}}(t)$. From these frames, the sequence $\hat{\underline{X}}(t)$ should be estimated. Note that some (if not all) frames will not have estimates for every pixel in $\hat{\underline{Z}}(t)$, necessitating a further joint interpolation and deblurring step, which will be discussed in Sections 3 and 4. For the cases of multiframe demosaicing and color SR, the above process is to be applied separately on the R, G, and B layers, producing the arrays we will start from in the next sections.

While the recursive procedure outlined above will produce the optimal (minimum mean-squared) estimate of the state (blurry image $\hat{\underline{Z}}(t)$) in a causal fashion, we can also consider the best estimate of the same given "all" the frames. This optimal estimate is obtained by a two-way recursive filtering operation known as "smoothing," which we discuss next.

2.3. Smoothing method

The fast and memory efficient data fusion method described above is suitable for causal, real-time processing, as it estimates the HR frames from the previously seen LR frames. However, oftentimes super-resolution is preformed offline, and therefore a more accurate estimate of an HR frame at a given time is possible by using both previous and future LR frames. In this section, we study such offline dynamic SR method also known as smoothed dynamic SR [20].

The smoothed data fusion method is a two-pass (forward-backward) algorithm. In the first pass, the LR frames pass through a forward data fusion algorithm similar to the method explained in Section 2.2, resulting in a set of HR estimates $\{\hat{\underline{Z}}(t)\}_{t=1}^N$ and their corresponding diagonal covariance matrices $\{\widehat{M}(t)\}_{t=1}^N$. The second pass runs backward

in time using those mean-covariance pairs, and improves these forward HR estimates, resulting in the smoothed mean-covariance pairs $\{\underline{\widehat{Z}}_s(t), \widehat{M}_s(t)\}_{t=1}^{N}$.

While it is possible to simply implement the second pass (backward estimation) similar to the forward KF algorithm, and obtain the smooth estimate by weighted averaging of the forward and backward estimates with respect to their covariance matrices, computationally more efficient methods are more desirable. We refer the reader to the appendix for a more detailed study of such algorithm based on the fixed-interval smoothing method of Rauch, Tung, and Striebel [21, 22].

3. DEBLURRING AND INTERPOLATION OF MONOCHROMATIC IMAGE SEQUENCES

To perform robust deblurring and interpolation, we use the MAP cost function

$$\epsilon\left(\underline{X}(t)\right) = \left|\left|A(t)\left(H\underline{X}(t) - \underline{\widehat{Z}}(t)\right)\right|\right|_2^2 + \lambda\Gamma\left(\underline{X}(t)\right), \tag{13}$$

and define our desired solution as

$$\underline{\widehat{X}}(t) = \underset{\underline{X}(t)}{\text{ArgMin}}\, \epsilon\left(\underline{X}(t)\right). \tag{14}$$

Here, the matrix $A(t)$ is a diagonal matrix whose values are chosen in relation to our confidence in the measurements that contributed to make each element of $\underline{\widehat{Z}}(t)$. These values have inverse relation to the corresponding elements in the matrix[2] $\widehat{M}(t)$. The regularization parameter, λ, is a scalar for properly weighting the first term (data fidelity cost) against the second term (regularization cost), and $\Gamma(\underline{X})$ is the regularization cost function. The regularization term provides some prior information about the solution of this ill-posed problem and stabilizes it, improves the rate of convergence, and helps remove artifacts. In this section, we propose regularization terms that yield good results for the case of monochromatic dynamic SR problem and in Section 4 we address proper regularization terms for color SR, and multiframe demosaicing problems.

For the case of monochromatic SR, many regularization terms have been proposed. Some have limited applications and are useful for some special types of images (e.g., application of maximum entropy type regularization terms [23] are generally limited to producing sharp reconstructions of point objects). Tikhonov [2, 4], total variation (TV) [24–26], and bilateral-total variation (BTV) [8] type regularization terms are more generally applicable. While implementation of Tikhonov prior usually results in images with smoothed edges, TV prior tends to preserve edges in reconstruction, as it does not severely penalize steep local gradients.

[2] Note that for the smoothed HR estimation cases, $\underline{\widehat{Z}}_s(t)$ and $\widehat{M}_s(t)$ substitute for $\underline{\widehat{Z}}(t)$ and $\widehat{M}(t)$.

Based on the spirit of TV criterion and a related technique called the bilateral filter [27, 28], the BTV regularization is computationally cheap to implement and effectively preserves edges (see [8] for a comparison of Tikhonov, total variation, and bilateral regularization cost functions). The bilateral-TV regularization term is defined as

$$\Gamma_{\text{BTV}}\left(\underline{X}(t)\right) = \sum_{l=-P}^{P}\sum_{m=-P}^{P} \alpha^{|m|+|l|}\left|\left|\underline{X}(t) - S_x^l S_y^m \underline{X}(t)\right|\right|_1. \tag{15}$$

S_x^l and S_y^m are the operators corresponding to shifting the image represented by $\underline{X}$ by l pixels in horizontal direction and m pixels in vertical direction, respectively. This cost function in effect computes derivatives across multiple resolution scales. The scalar weight, $0 < \alpha < 1$, is applied to give a spatially decaying effect to the summation of the regularization term. Note that image shifting and differencing operations are very cheap to implement.

The overall cost function is the summation of the data fidelity penalty term and the regularization penalty term:

$$\underline{\widehat{X}}(t) = \underset{\underline{X}(t)}{\text{ArgMin}}\left[\left|\left|A(t)\left(H\underline{X}(t) - \underline{\widehat{Z}}(t)\right)\right|\right|_2^2 + \lambda\sum_{l=-P}^{P}\sum_{m=-P}^{P}\alpha^{|m|+|l|}\left|\left|\underline{X}(t) - S_x^l S_y^m \underline{X}(t)\right|\right|_1\right]. \tag{16}$$

Steepest descent optimization may be applied to minimize this cost function, which can be expressed as

$$\underline{\widehat{X}}_{n+1}(t) = \underline{\widehat{X}}_n(t) + \beta\left\{H^T A^T(t)\left(A(t)H\underline{X}(t) - A(t)\underline{\widehat{Z}}(t)\right) + \lambda\sum_{l=-P}^{P}\sum_{m=-P}^{P}\alpha^{|m|+|l|}\left[I - S_y^{-m}S_x^{-l}\right] \times \text{sign}\left(\underline{X}(t) - S_x^l S_y^m \underline{X}(t)\right)\right\}, \tag{17}$$

where S_x^{-l} and S_y^{-m} define the transposes of matrices S_x^l and S_y^m, respectively, and have a shifting effect in the opposite directions as S_x^l and S_y^m, and β is the step size.

4. DEMOSAICING AND DEBLURRING OF COLOR (FILTERED) IMAGE SEQUENCES

Similar to what is described in Section 3, we deal with color sequences in a two-step process of image fusion and simultaneous deblurring and interpolation. In this section, first we describe the fundamentals of the multiframe demosaicing and color-SR problems (Section 4.1) and then describe the proposed method which results in optimal reconstruction of superresolved color images (Section 4.2).

4.1. Fundamentals of multiframe demosaicing and color SR

A color image is represented by combining three separate monochromatic images. Ideally, each pixel should correspond to three scalar values; one for each of the color bands (red, green, and blue). In practice, however, to reduce production cost, many digital cameras have only one color measurement per pixel. The detector array is a grid of CCDs, each made sensitive to one color by placing a color filter array (CFA) in front of the CCD. The Bayer pattern shown in Figure 3 (left) is a very common example of such a color filter. The values of missing color bands at every pixel are then synthesized using some form of interpolation from neighboring pixel values. This process is known as color demosaicing.

While numerous single-frame demosaicing methods have been proposed (see [29–37] as representative works), the reconstructed images are almost always contaminated with different amounts of color artifacts. This results from the ill-posed nature of the demosaicing problem. However, if multiple, spatially offset, color-filtered images of the same scene are available, one can combine them both to increase spatial resolution, and to produce a more effective overall demosaicing with significantly reduced artifacts. Such an approach may be termed multiframe demosaicing. What makes multiframe demosaicing challenging is that almost none of the single-frame demosaicing methods (but the very recent methods in [16, 17, 38, 39]) are directly applicable to it.

A related problem, color SR, addresses fusing a set of previously demosaiced color LR (or originally full color LR frames) to enhance their spatial resolution. To date, there is very little work addressing the problem of color SR. One possible solution involves applying monochromatic SR algorithms to each of the color channels independently [40, 41], while using the color information to improve the accuracy of motion estimation. Another approach is transforming the problem to a different color space, where chrominance layers are separated from luminance, and SR is applied only to the luminance channel [3]. Both of these methods are suboptimal as they do not fully exploit the correlation across the color bands.

In this section, we present a very efficient dynamic method applicable to multiframe demosaicing and also, to the standard color SR problems (where full RGB channels are already available). Referring to the mosaic effects, the geometries of the single-frame and multiframe demosaicing problems are fundamentally different, making it impossible to simply cross-apply traditional demosaicing algorithms to the multiframe situation. To better understand the multiframe demosaicing problem, we offer an example for the case of translational motion. Suppose that a set of color-filtered LR images is available (images on the left in Figure 3). We use the static two-step SR process explained in [16] to fuse these images. In the first step, LR images are upsampled, motion compensated, and averaged to result in what we call the static "shift-and-add" HR image.

The shift-and-add image on the right side of Figure 3 illustrates the pattern of sensor measurements in the HR image grid. In such situations, the sampling pattern is quite arbitrary depending on the relative motion of the LR images. This necessitates a different demosaicing algorithm than those designed for the original Bayer pattern.

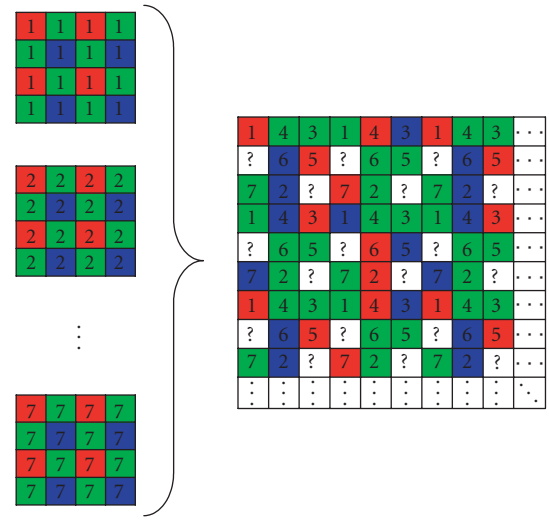


Figure 3: Fusion of 7 Bayer pattern LR images with relative translational motion (the figures in the left side of the accolade) results in an HR image ($\hat{Z}$) that does not follow Bayer pattern (the figure in the right side of the accolade). The symbol "?" represents the high-resolution pixel values that were undetermined after the shift-and-add step (result of insufficient LR frames).

Figure 3 shows that treating the green channel differently than the red or blue channels, as is done in many single-frame demosaicing methods before, is not particulary useful for the multiframe case. While globally there are more green pixels than blue or red pixels, locally any pixel may be surrounded by only red or blue colors. So, there is no general preference for one color band over the others.

Another assumption, the availability of one and only one color band value for each pixel, is also not correct in the multiframe case. In the underdetermined cases,[3] there are not enough measurements to fill the HR grid. The symbol "?" in Figure 3 represents such pixels. On the other hand, in the overdetermined case,[4] for some pixels, there may in fact be more than one color value available.

In the next subsection, we propose an algorithm for producing high-quality color sequences from a collection of LR color (filtered) images. Our computationally efficient MAP estimation method is motivated by the color image perception properties of the human visual system. This method is directly applicable to both color SR (given full RGB LR frames) and the more general multiframe demosaicing problems introduced earlier.

[3] Where the number of nonredundant LR frames is smaller than the square of resolution enhancement factor.

[4] Where the number of nonredundant LR frames is larger than the square of resolution enhancement factor.

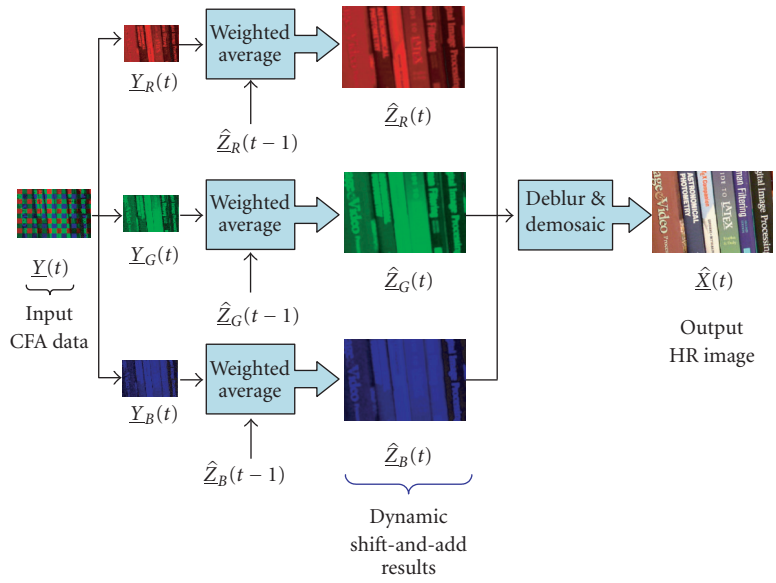


Figure 4: Block diagram representation of the overall dynamic SR process for color-filtered images. The feedback loops are omitted to simplify the diagram. Note that $\hat{\underline{Z}}_{i\in\{R,G,B\}}(t)$ represents the forward dynamic shift-and-add estimate studied in Section 2.2.

4.2. Multiframe demosaicing and color SR

As described in Section 2, our method is a two-step process of image fusion and simultaneous deblurring and interpolation. Figure 4 shows an overall block diagram of the dynamic SR process for mosaiced images (the feedback loops are eliminated to simplify the diagram). For the case of color SR, the first step involves nothing more than the application of the recursive image fusion algorithm separately on three different color bands. Image fusion of color-filtered images is done quite similarly, where each single-channel color-filtered frame is treated as a sparsely sampled three-channel color image. The second step (deblur and demosaic block in Figure 4) is the enhancement step that removes blur, noise, and color artifacts from the shift-and-add sequence, and is based on minimizing a MAP cost function with several terms composing an overall cost function similar to $\epsilon(\underline{X}(t))$ in (13). In what follows in this section, we define the terms in this cost function.

Data fidelity penalty term. This term penalizes the dissimilarity between the raw data and the HR estimate, and is defined as

$$J_0(\underline{X}(t)) = \sum_{i=R,G,B} ||A_i(t)(H\hat{\underline{X}}_i(t) - \hat{\underline{Z}}_i(t))||_2^2, \tag{18}$$

where $\hat{\underline{Z}}_R$, $\hat{\underline{Z}}_G$, and $\hat{\underline{Z}}_B$ are the three color channels of the color shift-and-add image, $\hat{\underline{Z}}$. A_R, A_G, and A_B are the red, green, and blue diagonal confidence matrices of $\hat{\underline{Z}}_R$, $\hat{\underline{Z}}_G$, and $\hat{\underline{Z}}_B$, respectively. The diagonal elements of $A_{i\in\{R,G,B\}}$ which correspond to those pixels of $\hat{Z}_{i\in\{R,G,B\}}$, which have not been produced from any measurement are set to zero. Note that the $A_{i\in\{R,G,B\}}$ matrices for the multiframe demosaicing problem are sparser than the corresponding matrices in the color SR case.

Luminance penalty term. The human eye is more sensitive to the details in the luminance component of an image than the details in the chrominance components [32]. Therefore, it is important that the edges in the luminance component of the reconstructed HR image look sharp. Applying bilateral-TV regularization to the luminance component will result in this desired property [8], where L_1 norm is used to force spatial smoothness while creating sharp edges. The luminance image can be calculated as the weighted sum $\underline{X}_L(t) = 0.299\underline{X}_R(t) + 0.597\underline{X}_G(t) + 0.114\underline{X}_B(t)$ as explained in [42]. The luminance regularization term is defined (as before):

$$J_1(\underline{X}(t)) = \sum_{l=-P}^{P}\sum_{m=-P}^{P} \alpha^{|m|+|l|} ||\underline{X}_L(t) - S_x^l S_y^m \underline{X}_L(t)||_1. \tag{19}$$

The S_x^l and S_y^m shifting operators and the parameter α are defined in (15).

Chrominance penalty term. The human eye is more sensitive to chromatic change in the low-spatial-frequency region than the luminance change [37]. As the human eye is less sensitive to the chrominance channel resolution, it can be smoothed more aggressively. Therefore, L_2 regularization is an appropriate method for smoothing the Chrominance term:

$$J_2(\underline{X}(t)) = ||\Lambda\underline{X}_{C1}(t)||_2^2 + ||\Lambda\underline{X}_{C2}(t)||_2^2, \tag{20}$$

where Λ is the matrix realization of a highpass operator such as the Laplacian filter. The images $\underline{X}_{C1}(t)$ and $\underline{X}_{C2}(t)$ are the I and Q layers in the YIQ color representation.

Orientation penalty term. This term penalizes the nonhomogeneity of the edge orientation across the color channels. Although different bands may have larger or smaller gradient magnitudes at a particular edge, the statistics of natural images shows that it is reasonable to assume a same edge orientation for all color channels. That is, for instance, if an edge appears in the red band at a particular location, then an edge with the same orientation should appear in the other color bands at the same location as well. Following [31], minimizing the vector product norm of any two adjacent color pixels forces different bands to have similar edge orientation. With some modifications to what was proposed in [31], our orientation penalty term is a differentiable cost function:

$$
\begin{aligned}
&J_3(\underline{X}(t)) \\
&= \sum_{l=-1}^{1}\sum_{m=-P}^{1}\Big[\left|\left|\underline{X}_G(t)\odot S_x^l S_y^m \underline{X}_B(t) - \underline{X}_B(t)\odot S_x^l S_y^m \underline{X}_G(t)\right|\right|_2^2 \\
&\qquad + \left|\left|\underline{X}_B(t)\odot S_x^l S_y^m \underline{X}_R(t) - \underline{X}_R(t)\odot S_x^l S_y^m \underline{X}_B(t)\right|\right|_2^2 \\
&\qquad + \left|\left|\underline{X}_R(t)\odot S_x^l S_y^m \underline{X}_G(t) - \underline{X}_G(t)\odot S_x^l S_y^m \underline{X}_R(t)\right|\right|_2^2\Big],
\end{aligned}
\tag{21}
$$

where $\odot$ is the element-by-element multiplication operator.

The overall cost function $\epsilon(\underline{X}(t))$ is the summation of these cost functions:

$$
\begin{aligned}
\hat{\underline{X}}(t) = \underset{\underline{X}(t)}{\text{ArgMin}}\,\big[&J_0(\underline{X}(t)) + \lambda' J_1(\underline{X}(t)) \\
&+ \lambda'' J_2(\underline{X}(t)) + \lambda''' J_3(\underline{X}(t))\big].
\end{aligned}
\tag{22}
$$

Coordinatewise steepest descent optimization may be applied to minimize this cost function. In the first step, the derivative of (22) with respect to one of the color bands is calculated, assuming the other two color bands are fixed. In the next steps, the derivative is computed with respect to the other color channels. The steepest descent iteration formulation for this cost function is shown in [17].

5. EXPERIMENTS

Experiments on synthetic and real data sets are presented in this section. In the first experiment, we synthesized a sequence of low-resolution color-filtered images from a single color image of size 1200×1600 captured with a one-CCD OLYMPUS C-4000 digital camera. A 128×128 section of this image was blurred with a symmetric Gaussian lowpass filter of size 4×4 pixels with standard deviation equal to one. The resulting images were subsampled by the factor of four in each direction and further color filtered with Bayer pattern creating a 32×32 image. We added Gaussian noise to the resulting LR frames to achieve SNR equal[5] to 30 dB. We consecutively shifted the 128×128 window on the original high-resolution image by one pixel in right, down, or up directions, and repeated the same image degradation process. In this fashion, we created a sequence of 250 frames.

FIGURE 5: A sequence of 250 LR color-filtered images is recursively fused (Section 2), increasing their resolution by the factor of 4 in each direction. They were further deblurred and demosaiced (Section 4), resulting in images with much higher quality than the input LR frames. In (a) and (b), we see the ground truth for frames #50 and #250, and (c) and (d) are the corresponding synthesized LR frames. In (e) and (f), we see the recursively fused HR frames, and (g) and (h) show the deblurred-demosaiced frames.

Figures 5(a) and 5(b) show two sections of the HR image. Figures 5(c) and 5(d) show frames #50 and #250 of the LR sequence (for the sake of presentation each frame has been demosaiced following the method of [29]). We created a sequence of HR fused images using the method described in Section 2.2 (factor of 4 resolution enhancement by forward shift-and-add method). Figures 5(e) and 5(f) show frames

[5] Signal-to-noise ratio (SNR) is defined as $10\log_{10}(\sigma^2/\sigma_n^2)$, where σ^2 and σ_n^2 are variances of a clean frame and noise, respectively.

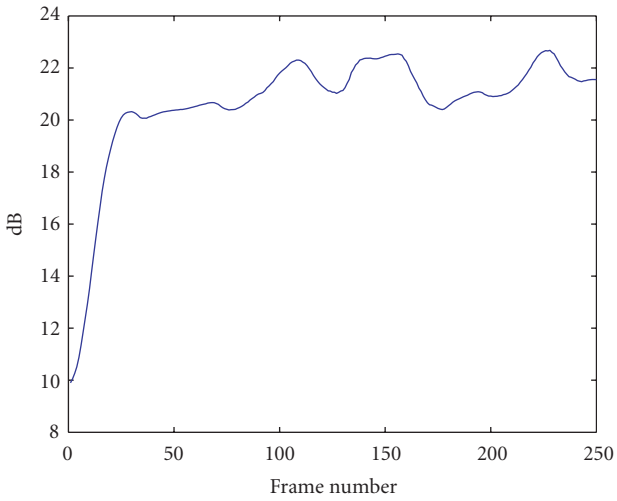


Figure 6: PSNR values in dB for the synthesized 250 frames sequence of the experiment in Figure 5.

#50 and #250 of this sequence, where the missing values were filled in using bilinear interpolation. Note that for the particular motion in this underdetermined experiment, it is easy to show that less than 1/3 of the pixel values in $\underline{\hat{Z}}(t)$ are determined by the shift-and-add process.

Later each frame was deblurred-demosaiced using the method described in Section 4. Figures 5(g) and 5(h) show frames #50 and #250 of this reconstructed sequence, where the color artifacts have been almost completely removed. The PSNR[6] values for this sequence are plotted in Figure 6. This plot shows that after the first few frames are processed, the quality of the reconstruction is stabilized for the remaining frames. The small distortions in the PSNR values of this sequence are due to the difference in color and high-frequency information of different frames. The corresponding parameters for this experiment (tuned by trial-and-error) were as follows: $\alpha = 0.9$, $\epsilon = 10^6$, $\beta = 0.06$, $\lambda' = \lambda'' = 0.001$, and $\lambda''' = 10$. Fifteen iterations of steepest descent were used for this experiment.

Our next experiment was preformed on a real-world (already demosaiced) compressed image sequence courtesy of Adyoron Intelligent Systems Ltd., Tel Aviv, Israel. Two frames of this sequence (frames #20 and #40) are shown in Figures 7(a) and 7(d). We created a sequence of HR fused images (factor of 4 resolution enhancement) using the forward data fusion method described in Section 2.2 (Figures 7(b) and 7(e)). Later each frame in this sequence was deblurred using the method described in Section 4 (Figures 5(c) and 7(f)). The corresponding parameters for this experiment are as follows: $\alpha = 0.9$, $\epsilon = 10^6$, $\beta = 0.1$, $\lambda' = \lambda'' = 0.005$, and $\lambda''' = 50$. Fifteen iterations of steepest descent were used for this experiment. The (unknown) camera PSF was assumed to be a 4×4 Gaussian kernel with standard deviation equal to one. As the relative motion between these images approximately followed the translational model, we only needed to estimate the motion between the luminance components of these images [43]. We used the method described in [44] to compute the motion vectors. In the reconstructed images, there are some effects of wrong motion estimation, seen as periodic teeth along the vertical bars. We assume that these errors correspond to the small deviations from the pure translational model.

In the third experiment, we used 74 uncompressed, raw CFA images from a video camera (based on Zoran 2MP CMOS sensors). We applied the method of [29] to demosaic each of these LR frames, individually. Figure 8(a) shows frame #1 of this sequence.

To increase the spatial resolution by a factor of three, we applied the proposed forward data fusion method of Section 2.2 on the raw CFA data. Figure 8(b) shows the forward shift-and-add result. This frame was further deblurred-demosaiced by the method explained in Section 4 and the result is shown in Figure 8(c). To enhance the quality of reconstruction, we applied the smoothing method of Section 2.3 to this sequence. Figure 8(d) shows the smoothed data fusion result for frame #1 (smoothed shift-and-add). The deblurred-demosaiced result of applying the method explained in Section 4 is shown in Figure 8(e).

Figure 8(f) shows the frame #69 of this sequence, demosaiced by the method in [29]. Figure 8(g) shows the result of applying the method of Section 2.3 to form the smoothed shift-and-add image. This frame is further deblurred-demosaiced by the method explained in Section 4 and the result is shown in Figure 8(h).

The parameters used for this experiment are as follows: $\beta = 0.04$, $\epsilon = 10^6$, $\alpha = 0.9$, $\lambda' = 0.001$, $\lambda'' = 50$, $\lambda''' = 0.1$. The (unknown) camera PSF was assumed to be a tapered 5×5 disk PSF.[7]

Note that $F(t)\underline{\hat{X}}(t-1)$ is a suitable candidate to initialize $\underline{\hat{X}}^0(t)$, since it follows the KF prediction of the state-vector updates. Therefore, as the deblurring-demosaicing step is the computationally expensive part of this algorithm, for all of these experiments we used the shifted version of deblurred image of $t-1$ as the initial estimate of the deblurred-demosaiced image at time instant t.

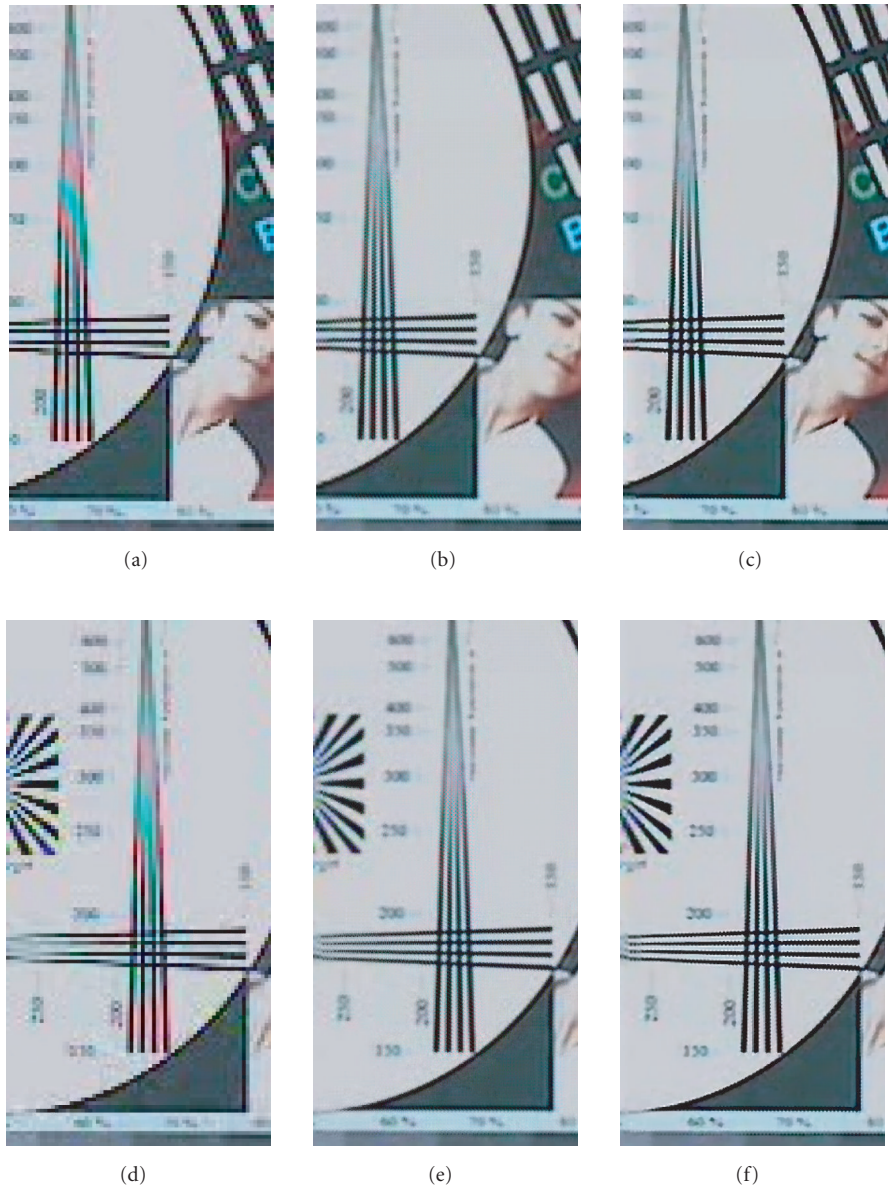
6. SUMMARY AND FUTURE WORK

In this paper, we presented algorithms to enhance the quality of a set of noisy, blurred, and possibly color-filtered images to produce a set of monochromatic or color HR images with less noise, aliasing, and blur effects. We used MAP estimation technique to derive a hybrid method of dynamic SR and multiframe demosaicing. Our method is also applicable to the case of color SR.

For the case of translational motion and common space-invariant motion, we justified a two-step algorithm. In the

[6] The PSNR of two vectors $\underline{X}$ and $\underline{\hat{X}}$ of size $[3r^2Q_1Q_2 \times 1]$ is defined as $\text{PSNR}(\underline{X}, \underline{\hat{X}}) = 10\log_{10}((255^2 \times 3r^2Q_1Q_2)/\|\underline{X} - \underline{\hat{X}}\|_2^2)$.

[7] Matlab command fspecial('disk',2) creates such a blurring kernel.



(a) (b) (c)

(d) (e) (f)

Figure 7: A sequence of 60 real-world LR compressed color frames ((a) and (d) show frames #20 and #40; resp.) is recursively fused (Section 2), increasing their resolution by the factor of four in each direction ((b) and (e), resp.). They were further deblurred (Section 4), resulting in images with much higher quality than the input LR frames ((c) and (f), resp.).

first step, we used the KF framework for fusing LR images recursively in a fast and memory-efficient way. In the second step, while deblurring and interpolating the missing values, we reduced luminance and color artifacts by using appropriate penalty terms. These terms were based on our prior knowledge of the statistics of natural images and the properties of the human visual system. All matrix-vector operations in the proposed method are implemented as simple image operators.

While the proposed demosaicing method is applicable to a very wide range of data and motion models, our dynamic SR method is developed for the case of translational motion and common space-invariant blur. A fast and robust recursive data fusion algorithm based on using L_1 norm minimization applicable to general motion models is part of our ongoing work.

APPENDIX

A. NONCAUSAL DYNAMIC SUPER-RESOLUTION

In this appendix, we explain and formulate the two-pass fixed-interval smoothing method of Rauch, Tung, and Striebel [21, 22] for the dynamic SR problem. The first pass is

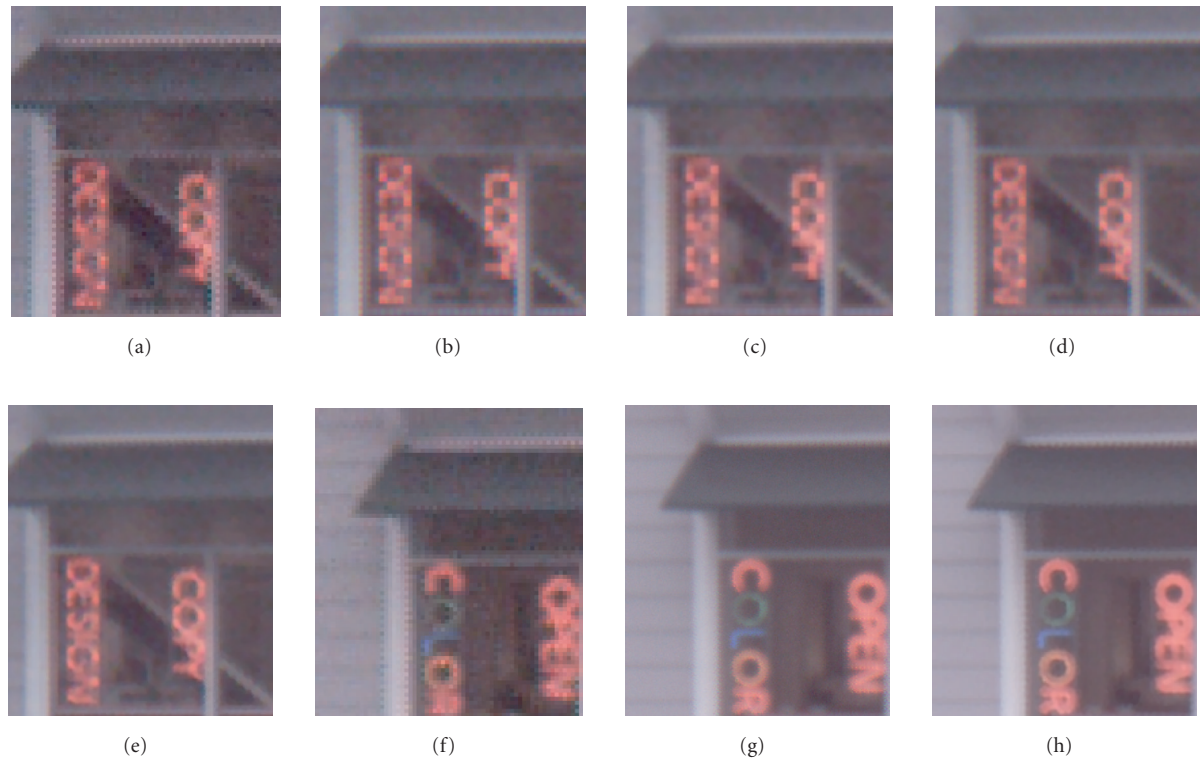


Figure 8: A sequence of 74 real-world LR uncompressed color-filtered frames ((a) and (f) show frames #1 and #69, resp.) is recursively fused (forward data fusion method of Section 2.2), increasing their resolution by the factor of three in each direction (b) and (g). They were further deblurred (Section 4), resulting in images with much higher quality than the input LR frames (c) and (h). The smoothed data fusion method of Section 2.3 further improves the quality of reconstruction. The smoothed shift-and-add result for frame #1 is shown in (d). This image was further deblurred-demosaiced (Section 4) and the result is shown in (e).

quite similar to the method explained in Section 2.2, resulting in a set of HR estimates $\{\hat{\underline{Z}}(t)\}_{t=1}^{N}$ and their corresponding diagonal covariance matrices $\{\widehat{M}(t)\}_{t=1}^{N}$. The second pass runs backward in time using those mean-covariance pairs, and improves these forward HR estimates.

The following equations define the HR image and covariance updates in the second pass. Assuming that we have the entire sequence $\{\hat{\underline{Z}}(t), \widehat{M}(t)\}_{t=1}^{N}$, we desire to estimate the pairs $\{\hat{\underline{Z}}_s(t), \widehat{M}_s(t)\}_{t=1}^{N}$ that represent the mean and covariance per time t, based on all the information in the sequence. We assume a process that runs from $t = N-1$ downwards, initialized with $\hat{\underline{Z}}_s(N) = \hat{\underline{Z}}(N)$ and $\widehat{M}_s(N) = \widehat{M}(N)$.

We start by the covariance propagation matrix. Notice its similarity to (5):

$$\widetilde{M}(t+1) = F(t+1)\widehat{M}(t)F^T(t+1) + C_v(t+1). \tag{A.1}$$

This equation builds a prediction of the covariance matrix for time $t+1$, based on the first-pass forward stage. Note that the outcome is diagonal as well.

The Kalman smoothed gain matrix is computed using the above prediction matrix, and the original forward covariance one, by

$$K_s(t) = \widehat{M}(t)F^T(t+1)\left[\widetilde{M}(t+1)\right]^{-1}. \tag{A.2}$$

This gain will be used both for the backward updates of the mean and the covariance,

$$\hat{\underline{Z}}_s(t) = \hat{\underline{Z}}(t) + K_s(t)\left[\hat{\underline{Z}}_s(t+1) - F(t+1)\hat{\underline{Z}}(t)\right], \tag{A.3}$$

where the term $\hat{\underline{Z}}_s(t+1) - F(t+1)\hat{\underline{Z}}(t)$ could be interpreted as a prediction error. The smoothed covariance matrix is updated by

$$\begin{aligned}\widehat{M}_s(t) &= \mathrm{Cov}\left(\hat{\underline{Z}}_s(t)\right)\\ &= \widehat{M}(t) + K_s(t)\left[\widehat{M}_s(t+1) - \widetilde{M}(t+1)\right]K_s^T(t).\end{aligned} \tag{A.4}$$

Following the notations we have used before, we use the superscript "b" to represent backward shifting in time of vectors and matrices, so that $\hat{\underline{Z}}_s^b(t) = F^T(t+1)\hat{\underline{Z}}_s(t+1)$ and similarly $\widehat{M}_s^b(t) = F^T(t+1)\widehat{M}_s(t+1)F(t+1)$ and $C_v^b(t) = F^T(t+1)C_v(t+1)F(t+1)$. Then, using the same rational practiced in the forward algorithm, the smoothed gain matrix for a pixel at spatial position q is

$$\frac{\left[\widehat{M}(t)\right]_q}{\left[\widehat{M}(t)\right]_q + \left[C_v^b(t)\right]_q}. \tag{A.5}$$

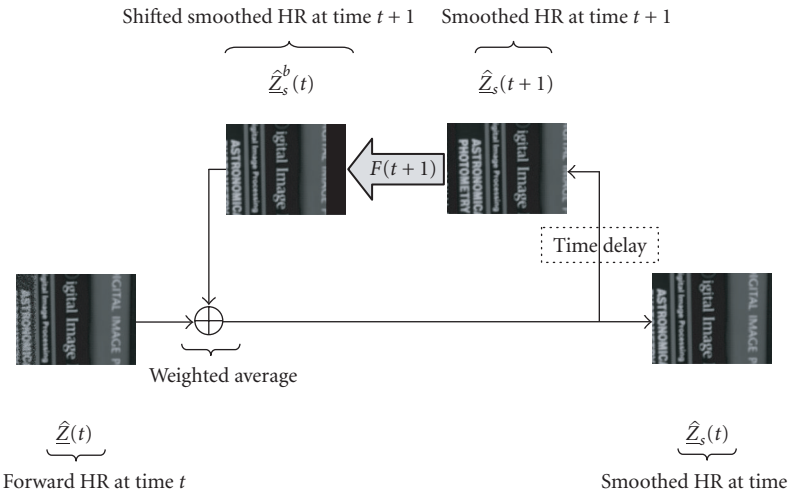


Figure 9: Block diagram representation of (A.7), where $\hat{\underline{Z}}_s(t)$, the new Rauch-Tung-Striebel smoothed HR output frame, is the weighted average of $\hat{\underline{Z}}(t)$, the forward Kalman HR estimate at time t, and $\hat{\underline{Z}}_s^b(t)$ is the previous smoothed estimate of the HR image($\hat{\underline{Z}}_s^b(t) = F^T(t+1)\hat{\underline{Z}}_s(t+1)$), after motion compensation.

(i) Task. Given $\{\underline{Y}(t)\}_{t\geq 1}$, estimate $\{\underline{Z}(t)\}_{t\geq 1}$ noncausally.
(ii) First Pass. Assume that the causal algorithm has been applied, giving the sequence $\{\hat{\underline{Z}}(t), \widehat{M}(t)\}_{t=1}^N$.
(iii) Initialization. Set $t = N$, choose $\hat{\underline{Z}}_s(t) = \hat{\underline{Z}}(t)$ and $\widehat{M}_s(t) = \widehat{M}(t)$.
(iv) Update process. Set $t \to t-1$ and apply
  (1) motion compensation: compute $\hat{\underline{Z}}_s^b(t) = F^T(t+1)\hat{\underline{Z}}_s(t+1)$ and $\widehat{M}_s^b(t) = F^T(t+1)\widehat{M}_s(t+1)F(t+1)$;
  (2) update of the covariance: use (A.6) to compute the update $\widehat{M}_s(t)$;
  (3) update of the mean: use (A.7) to compute the update $\hat{\underline{Z}}_s(t)$.
(v) Repeat. Update process.

Algorithm 2: Smoothed dynamic shift-and-add algorithm.

Similar to what is shown in Section 2.2, we can simplify (A.1), (A.2), (A.3), and (A.4) to the following pixelwise update formulas:

$$[\widehat{M}_s(t)]_q = [\widehat{M}(t)]_q + [\widehat{M}(t)]_q^2 \times \frac{[\widehat{M}_s^b(t)]_q - [\widehat{M}(t)]_q - [C_v^b(t)]_q}{[\widehat{M}(t)]_q + [C_v^b(t)]_q}, \tag{A.6}$$

$$[\hat{\underline{Z}}_s(t)]_q = \frac{[C_v^b(t)]_q[\hat{\underline{Z}}(t)]_q + [\widehat{M}(t)]_q[\hat{\underline{Z}}_s^b(t)]_q}{[\widehat{M}(t)]_q + [C_v^b(t)]_q}. \tag{A.7}$$

Figure 9 describes the above equation as a block diagram.

There is a simple interpretation for (A.7). The smoothed HR pixel at time t is the weighted average of the forward HR estimate at time t ($[\hat{\underline{Z}}(t)]_q$) and the smoothed HR pixel at time instant $t+1$ after motion compensation ($[\hat{\underline{Z}}_s^b(t)]_q$). In case there is high confidence in the $[\hat{\underline{Z}}(t)]_q$ (i.e., the value of $[\widehat{M}(t)]_q$ is small), the weight of $[\hat{\underline{Z}}_s^b(t)]_q$ will be small. On the other hand, if there is high confidence in estimating the HR pixel at time $t+1$ from an HR pixel at time t after proper motion compensation (i.e., the value of $[C_v^b(t)]_q$ is small), it is reasonable to assume that the smoothed HR pixel at time t can be estimated from a HR pixel at time $t+1$ after proper motion compensation. Note that unlike the forward pass, estimation of HR smoothed images do not depend on the computation of smoothed covariance update matrices as in (A.4) and (A.6), and those can be ignored in the application.

The overall procedure using these update equations is outlined in Algorithm 2.

ACKNOWLEDGMENTS

We would like to thank the associate editor and the reviewers for valuable comments that helped improving the clarity of presentation of this paper, and Lior Zimet and Erez Galil from Zoran Corp. for providing the camera used to produce the raw CFA images of experiment 3 in Figure 8. This work was supported in part by the US Air Force Grant F49620-03-1-0387, and by the National Science Foundation, Science and Technology Center for Adaptive Optics, managed by the University of California at Santa Cruz under Cooperative Agreement no. AST-9876783. M. Elad's work was supported in part by the Jewish Communities of Germany Research Fund.

REFERENCES

[1] T. S. Huang and R. Y. Tsai, "Multi-frame image restoration and registration," in *Advances in Computer Vision and Image Processing*, vol. 1, chapter 7, pp. 317–339, JAI Press, Greenwich, Conn, USA, 1984.

[2] N. Nguyen, P. Milanfar, and G. H. Golub, "A computationally efficient super-resolution image reconstruction algorithm," *Transactions on Image Processing*, vol. 10, no. 4, pp. 573–583, 2001.

[3] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 3, pp. 231–239, 1991.

[4] M. Elad and A. Feuer, "Restoration of a single super-resolution image from several blurred, noisy, and undersampled measured images," *Transactions on Image Processing*, vol. 6, no. 12, pp. 1646–1658, 1997.

[5] A. Zomet and S. Peleg, "Efficient super-resolution and applications to mosaics," in *Proceedings of IEEE 15th International Conference on Pattern Recognition (ICPR '00)*, vol. 1, pp. 579–583, Barcelona, Spain, September 2000.

[6] M. K. Ng and N. K. Bose, "Mathematical analysis of super-resolution methodology," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 62–74, 2003.

[7] Y. Altunbasak, A. J. Patti, and R. M. Mersereau, "Super-resolution still and video reconstruction from MPEG-coded video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 4, pp. 217–226, 2002.

[8] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multi-frame super-resolution," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1327–1344, 2004.

[9] C. A. Segall, A. K. Katsaggelos, R. Molina, and J. Mateos, "Bayesian resolution enhancement of compressed video," *IEEE Transactions on Image Processing*, vol. 13, no. 7, pp. 898–911, 2004.

[10] S. Borman and R. L. Stevenson, "Super-resolution from image sequences—a review," in *Proceedings of Midwest Symposium on Circuits and Systems (MWSCAS '98)*, pp. 374–378, Notre Dame, Ind, USA, August 1998.

[11] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[12] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *International Journal of Imaging Systems and Technology*, vol. 14, no. 2, pp. 47–57, 2004.

[13] M. Elad and A. Feuer, "Super-resolution restoration of an image sequence: adaptive filtering approach," *IEEE Transactions on Image Processing*, vol. 8, no. 3, pp. 387–395, 1999.

[14] M. Elad and A. Feuer, "Super-resolution reconstruction of image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 817–834, 1999.

[15] M. Elad and Y. Hel-Or, "A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur," *IEEE Transactions on Image Processing*, vol. 10, no. 8, pp. 1187–1193, 2001.

[16] S. Farsiu, M. Elad, and P. Milanfar, "Multiframe demosaicing and super-resolution from undersampled color images," in *Computational Imaging II*, vol. 5299 of *Proceedings of SPIE*, pp. 222–233, San Jose, Calif, USA, January 2004.

[17] S. Farsiu, M. Elad, and P. Milanfar, "Multiframe demosaicing and super-resolution of color images," *IEEE Transactions on Image Processing*, vol. 15, no. 1, pp. 141–159, 2006.

[18] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.

[19] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, New York, NY, USA, 1970.

[20] M. Elad, *Super-resolution reconstruction of continuous image sequence*, Ph.D. dissertation, Technion-Israel Institute of Technology, Haifa, Israel, 1997.

[21] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum likelihood estimates of dynamic linear systems," *American Institute of Aeronautics and Astronautics*, vol. 3, no. 8, pp. 1445–1450, 1965.

[22] A. C. Harvey, *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge University Press, Cambridge, UK, 1990.

[23] A. Bovik, *Handbook of Image and Video Processing*, Academic Press, New York, NY, USA, 2000.

[24] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1–4, pp. 259–268, 1992.

[25] Y. Li and F. Santosa, "A computational algorithm for minimizing total variation in image restoration," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 987–995, 1996.

[26] T. F. Chan, S. Osher, and J. Shen, "The digital TV filter and nonlinear denoising," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 231–241, 2001.

[27] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of IEEE 6th International Conference on Computer Vision (ICCV '98)*, pp. 839–846, Bombay, India, January 1998.

[28] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Transactions on Image Processing*, vol. 11, no. 10, pp. 1141–1151, 2002.

[29] C. Laroche and M. Prescott, "Apparatus and method for adaptive for adaptively interpolating a full color image utilizing chrominance gradients," United States Patent 5,373,322, 1994.

[30] R. Kimmel, "Demosaicing: image reconstruction from color CCD samples," *IEEE Transactions on Image Processing*, vol. 8, no. 9, pp. 1221–1228, 1999.

[31] D. Keren and M. Osadchy, "Restoring subsampled color images," *Machine Vision and Applications*, vol. 11, no. 4, pp. 197–202, 1999.

[32] Y. Hel-Or and D. Keren, "Demosaicing of color images using steerable wavelets," Tech. Rep. HPL-2002-206R1 20020830,

HP Laboratories Israel, Haifa, Israel, 2002, Online, available: citeseer.nj.nec.com/.

[33] D. D. Muresan and T. W. Parks, "Optimal recovery demosaicing," in *Proceedings of IASTED International Conference on Signal and Image Processing (SIP '02)*, Kauai, Hawaii, USA, August 2002.

[34] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 997–1013, 2002.

[35] D. Alleysson, S. Süsstrunk, and J. Hérault, "Color demosaicing by estimating luminance and opponent chromatic signals in the Fourier domain," in *Proceedings of IS&T/SID 10th Color Imaging Conference*, pp. 331–336, Scottsdale, Ariz, USA, November 2002.

[36] R. Ramanath, W. E. Snyder, G. L. Bilbro, and W. A. Sander, "Demosaicking methods for the Bayer color arrays," *Journal of Electronic Imaging*, vol. 11, no. 3, pp. 306–315, 2002.

[37] S.-C. Pei and I.-K. Tam, "Effective color interpolation in CCD color filter arrays using signal correlation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 6, pp. 503–513, 2003.

[38] A. Zomet and S. Peleg, "Multi-sensor super-resolution," in *Proceedings of IEEE 6th Workshop on Applications of Computer Vision (WACV '02)*, pp. 27–31, Orlando, Fla, USA, December 2002.

[39] T. Gotoh and M. Okutomi, "Direct super-resolution and registration using raw CFA images," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Patern Recognition (CVPR '04)*, vol. 2, pp. 600–607, Washington, DC, USA, June–July 2004.

[40] N. R. Shah and A. Zakhor, "Resolution enhancement of color video sequences," *IEEE Transactions on Image Processing*, vol. 8, no. 6, pp. 879–885, 1999.

[41] B. C. Tom and A. K. Katsaggelos, "Resolution enhancement of monochrome and color video using motion compensation," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 278–287, 2001.

[42] W. K. Pratt, *Digital Image Processing*, John Wiley & Sons, New York, NY, USA, 3rd edition, 2001.

[43] P. Golland and A. M. Bruckstein, "Motion from color," *Computer Vision and Image Understanding*, vol. 68, no. 3, pp. 346–362, 1997.

[44] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Proceedings of European Conference on Computer Vision (ECCV '92)*, pp. 237–252, Santa Margherita Ligure, Italy, May 1992.

Sina Farsiu received the B.S. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 1999 and the M.S. degree in biomedical engineering from the University of Tehran, Tehran, in 2001. He is currently pursuing the Ph.D. degree in electrical engineering at the University of California, Santa Cruz. His technical interests include signal and image processing, adaptive optics, and artificial intelligence.

Michael Elad received the B.S., M.S., and D.S. degrees from the Department of Electrical Engineering at Technion – Israel Institute of Technology (IIT), Haifa, Israel, in 1986, 1988, and 1997, respectively. From 1988 to 1993, he served in the Israeli Air Force. From 1997 to 2000, he worked at Hewlett-Packard Laboratories as an R&D Engineer. From 2000 to 2001, he headed the research division at Jigami Corporation, Israel. From 2001 to 2003, he was a Research Associate with the Computer Science Department, Stanford University (SCCM program), Stanford, Calif. In September 2003, he joined the Department of Computer Science, IIT, as an Assistant Professor. He was also a Research Associate at IIT from 1998 to 2000, teaching courses in the Electrical Engineering Department. He works in the field of signal and image processing, specializing, in particular, on inverse problems, sparse representations, and overcomplete transforms. Dr. Elad received the Best Lecturer Award twice (in 1999 and 2000). He is also the recipient of the Guttwirth and the Wolf Fellowships.

Peyman Milanfar received the B.S. degree in electrical engineering and mathematics from the University of California, Berkeley, and the S.M., E.E., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1988, 1990, 1992, and 1993, respectively. Until 1999, he was a Senior Research Engineer at SRI International, Menlo Park, Calif. He is currently an Associate Professor of electrical engineering, University of California, Santa Cruz. He was a Consulting Assistant Professor of computer science at Stanford University, Stanford, Calif, from 1998 to 2000, where he was also a Visiting Associate Professor from June to December 2002. His technical interests are in statistical signal and image processing and inverse problems. Dr. Milanfar won a National Science Foundation Career Award in 2000 and he was an Associate Editor for the IEEE Signal Processing Letters from 1998 to 2001.