

## University of Dayton eCommons

---

Electrical and Computer Engineering Faculty  
Publications

Department of Electrical and Computer  
Engineering

---


12-2007

# A Fast Image Super-Resolution Algorithm Using an Adaptive Wiener Filter

Russell C. Hardie

University of Dayton, [rhodie1@udayton.edu](mailto:rhodie1@udayton.edu)

Follow this and additional works at: [https://ecommons.udayton.edu/ece\\_fac\\_pub](https://ecommons.udayton.edu/ece_fac_pub)

 Part of the [Optics Commons](#), [Other Electrical and Computer Engineering Commons](#), [Other Physics Commons](#), and the [Signal Processing Commons](#)

---

### eCommons Citation

Hardie, Russell C., "A Fast Image Super-Resolution Algorithm Using an Adaptive Wiener Filter" (2007). *Electrical and Computer Engineering Faculty Publications*. 66.

[https://ecommons.udayton.edu/ece\\_fac\\_pub/66](https://ecommons.udayton.edu/ece_fac_pub/66)

This Article is brought to you for free and open access by the Department of Electrical and Computer Engineering at eCommons. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Publications by an authorized administrator of eCommons. For more information, please contact [frice1@udayton.edu](mailto:frice1@udayton.edu), [mschlangen1@udayton.edu](mailto:mschlangen1@udayton.edu).

# A Fast Image Super-Resolution Algorithm Using an Adaptive Wiener Filter

Russell C. Hardie, *Senior Member, IEEE*

## Abstract

A computationally simple super-resolution algorithm using a type of adaptive Wiener filter is proposed. The algorithm produces an improved resolution image from a sequence of low-resolution (LR) video frames with overlapping field of view. The algorithm uses subpixel registration to position each LR pixel value on a common spatial grid that is referenced to the average position of the input frames. The positions of the LR pixels are not quantized to a finite grid as with some previous techniques. The output high-resolution (HR) pixels are obtained using a weighted sum of LR pixels in a local moving window. Using a statistical model, the weights for each HR pixel are designed to minimize the mean squared error and they depend on the relative positions of the surrounding LR pixels. Thus, these weights adapt spatially and temporally to changing distributions of LR pixels due to varying motion. Both a global and spatially varying statistical model are considered here. Since the weights adapt with distribution of LR pixels, it is quite robust and will not become unstable when an unfavorable distribution of LR pixels is observed. For translational motion, the algorithm has a low computational complexity and may be readily suitable for real-time and/or near real-time processing applications. With other motion models, the computational complexity goes up significantly. However, regardless of the motion model, the algorithm lends itself to parallel implementation. The efficacy of the proposed algorithm is demonstrated here in a number of experimental results using simulated and real video sequences. A computational analysis is also presented.

EDICS : ISR-SUPR – Interpolation and Super-Resolution: Super-resolution

## Index Terms

super-resolution, image restoration, aliasing, undersampling, video processing

Manuscript submitted May 2007, revised August 2007, Sept. 2007.

Russell C. Hardie is with the Department of Electrical and Computer Engineering, University of Dayton, 300 College Park, Dayton, OH, 45469-0226, Phone: (937) 229-3611, Fax: (937) 229-2097, (Email: rhardie@udayton.edu).

# A Fast Image Super-Resolution Algorithm Using an Adaptive Wiener Filter

## I. INTRODUCTION

During acquisition, digital images are invariably degraded by a number of phenomena that limit image resolution and utility. Factors that can limit the effective resolution of an imaging system may include aliasing due to undersampling, blur, and noise. In many imaging applications, there is a desire to capture a wide field-of-view image with high spatial resolution. This generally leads to the selection of small f-number optics. For diffraction limited optics, the spatial cut-off frequency of the image in the focal plane is inversely proportional to the f-number [1]. Thus, the detector pitch of the focal plane array (FPA), that samples the image formed by the optics, must be properly matched with the f-number of the optics [1]. In particular, small f-number optics requires an FPA with small detector pitch. However, small detector pitch implies small detectors and these may be starved for photons due to the limited active surface area. Furthermore, sufficiently small detectors may be costly or impractical to produce. For these reasons, many imaging systems are designed to allow some aliasing in an engineering tradeoff.

When aliasing is a significant factor, multi-frame super-resolution (SR) enhancement has proven to be an effective post-processing method to help overcome resolution limitations stemming from non-ideal sampling from the FPA [2]. Multi-frame SR algorithms use a set of undersampled low-resolution (LR) frames from a video sequence to produce a high-resolution (HR) image. If some type of sub-pixel motion exists between frames, a unique set of samples is provided by the individual frames. This allows post-processing techniques to exploit the diversity of scene samples in multiple frames to effectively increase the sampling rate of a given FPA. The SR algorithms also typically seek to reduce blur from spatial detector integration, diffraction from the optics, and other sources.

A variety of SR algorithms have been proposed and these include frequency domain approaches, iterative HR image reconstruction techniques, learning-based methods and interpolation-restoration approaches [2]. The simplest of these SR methods, both computationally and intuitively, are the interpolation-restoration approaches [3–17]. In interpolation-restoration methods, registration is used to position the LR pixels from multiple frames onto a common HR grid. Since these LR pixels will generally lie nonuniformly on the common grid, a nonuniform interpolation scheme is typically employed to create uniformly spaced pixel values. This creates an image with an effectively reduced detector pitch, but

the spatial integration from the original detector size remains along with blur from other factors such as diffraction from the optics. For these reasons, the nonuniform interpolation is typically followed by a restoration method to treat the effects of the system point spread function (PSF). Most of these interpolation-restoration SR algorithms completely de-couple the interpolation and restoration steps by treating them independently. However, when the motion for a given set of frames is poorly distributed, any nonuniform interpolation result will undoubtedly suffer due to aliasing. An independent deconvolution process can exaggerate these artifacts. Therefore, integrating the nonuniform interpolation and deconvolution steps can be advantageous.

The partition weighted sum (PWS) SR algorithm in [17] is one interpolation-restoration approach that does combine the interpolation and restoration steps. However, in the PWS SR algorithm, the LR pixels are placed into a discrete HR grid by rounding off the coordinates of the LR pixels and placing them into the nearest HR grid position. While this is necessary for the PWS SR filter because of how the partition statistics are estimated, it does effectively introduce a quantization error in the registration that can affect performance. An off-line training procedure is used to obtain the filter weights for the PWS SR filter using training images. The method in [17] has the option to employ a bank of filters [18–20] tuned to different spatial structures in the image based on vector quantization. Another interpolation-restoration SR approach that seeks to combine the interpolation and restoration steps is the small-kernel Wiener filter (SKWF) in [16]. This method applies a space-invariant linear filter to the nonuniform samples on the HR grid. The filter impulse response is computed using a frequency domain analysis relying on the shift property of the Fourier transform. As such, it does not readily extend (in an optimum way) to rotation or other motion models.

In this paper, we propose a new computationally simple SR algorithm using a type of adaptive Wiener filter (AWF). The proposed AWF SR method combines nonuniform interpolation and restoration into a single weighted sum operation. In particular, the AWF SR method populates a common high resolution grid using the LR frames and the registration information. Unlike that in [17], the proposed method does not require quantization of the LR pixel coordinates to populate a discrete HR grid. Here we make use of the exact LR pixel locations provided by the registration algorithm. The AWF SR algorithm then produces the HR pixel estimates using weighted sums of neighboring LR pixels. The weights for each HR pixel are designed to minimize mean squared error based on the relative spatial locations of the LR pixels. In addition to avoiding the quantization of the LR pixel locations, another major difference between the proposed method and that in [17] is that here we use a parametric statistical model for the correlations that ultimately define the filter weights. It is this parametric model that allows the algorithm to

handle non-quantized motion parameters. It also gives the system a small number of tuning parameters to control performance and eliminates the need for empirical training images and the computational burden of estimating statistics from those images.

Another novel aspect of this work is that we apply a spatially varying statistical model in the spirit of [21,22] to control the SR filter weights. Thus, the filter weights adapt spatially and temporally to changing spatial distributions of LR pixels on the HR grid, and to the local intensity statistics of those LR pixels. Since the weights adapt with distribution of LR pixels, it is quite robust and will not become unstable when an unfavorable distribution of LR pixels is observed. The proposed spatially-adaptive algorithm is simpler computationally for both training and filtering than the vector quantization approach of the PWS SR method [17]. Note that with or without the spatially varying statistical model, a potentially distinct and optimum set of weights is computed for estimating each HR pixel on the HR grid. This is because the spatial location of the LR pixel around each HR pixel is considered individually, allowing the filter weights to be optimized for each HR pixel independently. This gives the proposed system more degrees of freedom (in the form of distinct filter weights) than the spatially invariant linear filter employed in [16].

Although we focus on global translational motion, the proposed system can also be applied to other motion models. We show that for global translational motion, the algorithm has a low computational complexity and may be readily suitable for real-time and/or near real-time processing applications. With other motion models, the computational complexity goes up significantly. However, regardless of the motion model, the algorithm lends itself to a parallel implementation. Thus, it may be possible to implement the non-translational version in real-time or near real-time using a massively parallel hardware architecture.

The remainder of the paper is organized as follows. Section II describes the observation model used as the basis for the proposed algorithm development. This observation model relates the ideal continuous scene to the observed LR pixels. The proposed AWF SR algorithm is presented in Section III. An analysis of computational complexity is provided in Section IV and experimental results are provided in Section V. Finally, conclusions are presented in Section VI.

## II. OBSERVATION MODEL

The proposed SR algorithm is based on the observation model shown in Figure 1. This block diagram illustrates how the LR frames are formed from the desired 2d continuous scene and how they relate to the corresponding ideal HR image. The desired scene is shown in Fig. 1 as  $d(x, y)$ , where  $x$  and  $y$  are the continuous spatial coordinates. This scene first goes through a geometric transformation to account

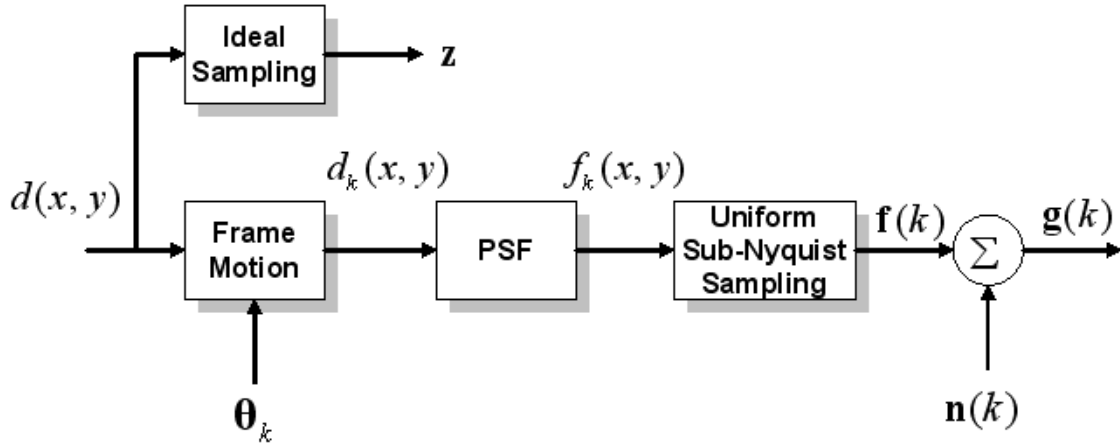


Fig. 1. Observation model relating a desired 2d continuous scene,  $d(x, y)$ , with a set of corresponding LR frames.

for any scene motion exhibited during the acquisition of  $P$  frames, yielding

$$d_k(x, y) = T_{\theta_k} \{d(x, y)\}, \quad (1)$$

for  $k = 1, 2, \dots, P$ , where  $\theta_k$  are the motion parameters associated with frame  $k$ . Commonly employed motion models in SR algorithms are translation, rotation, and affine transformations. Some previous work has considered non-global motion [23] and general optical flow. However, the needed subpixel accuracy is difficult to achieve when a large number of motion parameters are being estimated with a limited number of pixels. Here we focus primarily on translational motion, but the proposed algorithm is suitable for rotational motion when the PSF is approximately circularly symmetric, as will be described below. It may also yield useful results for other types of motion.

After the geometric transformation, the image is convolved with the system PSF. The PSF can be designed to include diffraction from the optics, spatial detector integration, and possibly other factors such as motion during integration and atmospheric turbulence. The latter two are generally time varying and are not considered here. Note that if the system is highly undersampled, the contribution of the detector spatial integration is generally far more dominant than the diffraction component. This allows for a relatively simple PSF model determined by the shape of the active area of a detector in the FPA [1]. The convolution operation is denoted

$$f_k(x, y) = d_k(x, y) * h(x, y), \quad (2)$$

where  $h(x, y)$  is the system PSF. Modeling the system PSF for undersampled systems is addressed in [1]

and we refer the reader to that paper for more details. However, an essential point to make here is that the system PSF bandlimits the image in the focal plane, making it possible to sample at the Nyquist rate with finite detector spacing. Note that the radial cut-off frequency associated with PSF from diffraction-limited optics with a circular exit pupil is [24]

$$\rho_c = \frac{1}{\lambda \mathcal{N}}, \quad (3)$$

where  $\mathcal{N}$  is the f-number of the optics and  $\lambda$  is the wavelength of light considered. Thus, the detector pitch in the FPA,  $\delta_s$ , would need to meet the requirement

$$\delta_s \leq \frac{1}{2\rho_c} = \frac{\lambda \mathcal{N}}{2} \quad (4)$$

to prevent aliasing.

The next step in the observation model is to spatially sample the image based on the detector layout and pitch in the FPA. This gives rise to a set of sampled scene values for each frame. In lexicographical form, we denote the samples for frame  $k$  with the vector  $\mathbf{f}(k)$ . As discussed in Section I, the system is likely to be undersampled with respect to the Nyquist criterion. Finally, we use an additive noise model to account for electronic and photometric noise sources. This yields the set of measured pixel values for frame  $k$ , denoted  $\mathbf{g}(k)$ . Note that all of the LR pixels values can be combined into a single vector yielding  $\mathbf{g} = [\mathbf{g}(1)^T, \mathbf{g}(2)^T, \dots, \mathbf{g}(P)^T]^T$ . On the other hand, the ideal HR image would be obtained by sampling  $d(x, y)$  at the Nyquist rate, with  $d(x, y)$  first bandlimited to  $\rho_c$ . Let this ideal HR image be represented in lexicographical notation with the  $N \times 1$  vector  $\mathbf{z}$  as shown in Fig. 1, where  $N$  is the number of HR pixels. This ideal HR image enjoys the full resolution afforded by the optics, with no aliasing, blur, or noise. It is this image we wish to estimate from the observation  $\mathbf{g}$ .

While the model shown in Fig. 1 is a good representation of the physical image acquisition process, a more convenient model for the following algorithm development is shown in Fig. 2. Here, the geometric transformation of the motion model is removed and replaced with a nonuniform sampling process guided by the motion parameters. Note that the models in Figs. 1 and 2 are equivalent for translational motion. This is because, for translational motion, the PSF blurring and the geometric transformation commute [13] (i.e., shifting then blurring is equivalent to blurring then shifting). It can be shown that the models are also equivalent for rotational motion, if the PSF is circularly symmetric. This follows because the continuous operations of rotation and convolution with a circularly symmetric PSF commute as well (i.e., rotating then blurring with a circularly symmetric PSF is equivalent to blurring then rotating). Once the PSF and motion models are swapped, the motion and uniform sampling operations in Fig. 1 can be combined into a single nonuniform sampling operation as shown in Fig. 2. Thus, sample locations from the image

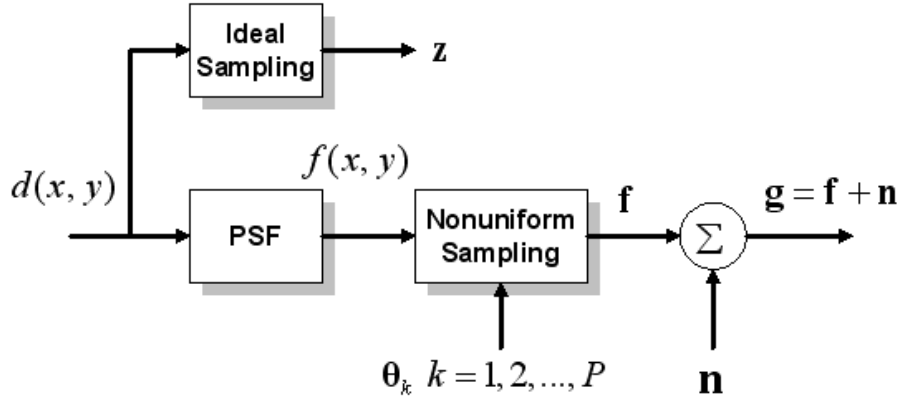


Fig. 2. Alternative observation model replacing the motion model and combining of LR frames by a single non-uniform sampling process. This is equivalent to the observation model in Fig. 1 for translational motion. It is also equivalent for rotational motion if the PSF is circularly symmetric.

$f(x, y)$  are obtained based on the registration parameters. This is done such that we end up with same set of samples in  $\mathbf{g}$  as those obtained by combining the outputs in Fig. 1. In the following section, we use the observation model in Fig. 2 to develop the proposed AWF SR algorithm.

### III. PROPOSED AWF SR ALGORITHM

The goal of the proposed AWF SR algorithm is to use  $\mathbf{g}(k)$ , for  $k = 1, 2, \dots, P$  to form an estimate of the HR image  $\mathbf{z}$ , denoted  $\hat{\mathbf{z}}$ . The effective sampling rate for the estimated image is to be greater than that in the observation model in Fig. 1 by  $L_x$  in the horizontal direction and  $L_y$  in the vertical direction. Ideally,  $L_x$  and  $L_y$  would be chosen so that the new SR sampling meets or exceeds the Nyquist criterion assuming a cutoff frequency of  $\rho_c$  given in (3). Note that the proposed system can be used to process video sequences by using a sliding window of input frames and outputting one HR image for each group of  $P$  input frames. Of course, it can also be used to generate a single HR still image from an input sequence.

#### A. System Overview

An overview of the AWF SR algorithm is provided by the block diagram in Fig. 3. The first step is to register the frames to a common grid. Here we employ the gradient-based registration technique described in [1, 25, 26]. To minimize the impact of noise and aliasing on the registration process, we



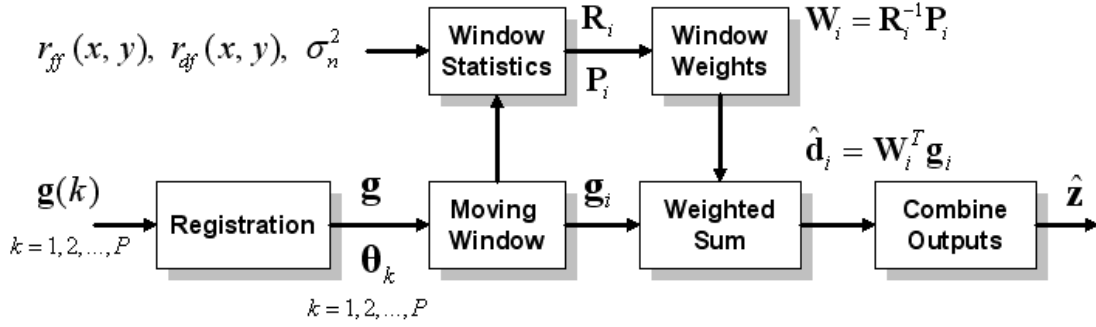


Fig. 3. Overview of the proposed SR algorithm.

employ a Gaussian prefilter, with a standard deviation of 1.5 LR pixel spacings, to the input images. Since this method works best for small shifts and rotations, we register each frame to the previous frame and then accumulate these. The HR grid is aligned with the average frame position, which provides motion stabilization for video processing with no added computational complexity. The resulting motion parameters are stored in  $\theta_k$ , for  $k = 1, 2, \dots, P$ . With these parameters, it is possible to establish the coordinates of each LR pixel in  $\mathbf{g}$  on a common grid. As mentioned in Section I, here we do not quantize the shifts and force the LR pixels into a discrete HR grid as done in [17]. This gives the current algorithm an advantage, especially for small upsampling factors where such quantization is significant.

Next, we employ a moving observation window on the HR grid as shown in Fig. 4. This window spans  $W_x$  HR pixel spacings in the horizontal direction and  $W_y$  HR pixel spacings in the vertical direction. All of the LR pixels on the HR grid that lie within the span of this observation window are placed into an observation vector  $\mathbf{g}_i = [g_{i,1}, g_{i,2}, \dots, g_{i,K_i}]^T$ , where  $i$  is a positional index for the window on the HR grid and  $K_i$  is the number of LR pixels within the  $i$ 'th observation window. Consider the case of translational motion when  $W_x$  and  $W_y$  are integer multiples of  $L_x$  and  $L_y$ , respectively. In this case, the number of LR pixels contained in each observation vector is constant and given by

$$K = \frac{PW_x W_y}{L_x L_y}. \quad (5)$$

For each observation window, we form estimates for the HR pixels within a subwindow of the main observation window, as shown in Fig. 4. Let the size of this subwindow be denoted  $D_x \times D_y$ , where  $1 \leq D_x \leq W_x$  and  $1 \leq D_y \leq W_y$ . Note that the estimation subwindow depicted in Fig. 4 is of size  $D_x = D_y = L_x = L_y = 6$ . The HR pixel estimates within the estimation subwindow are obtained using

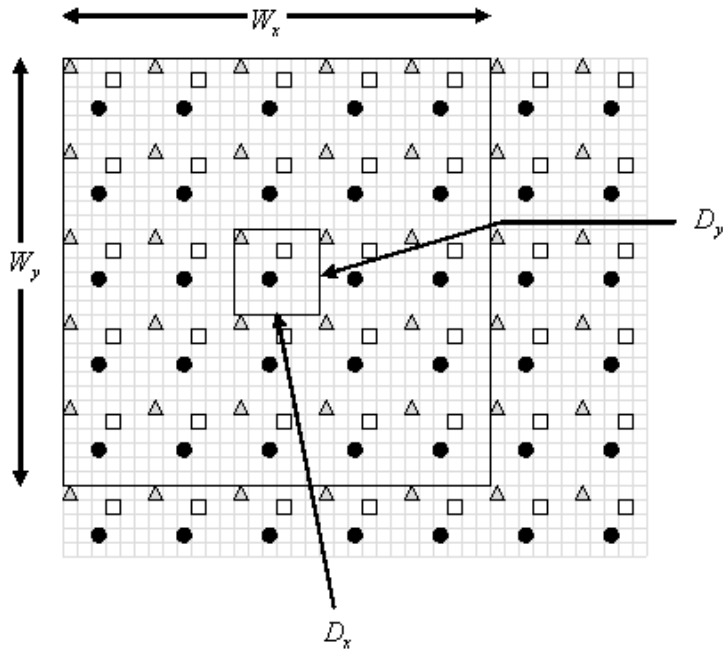


Fig. 4. High resolution grid showing the LR pixel locations for three frames (circles, triangles, and squares). The large box represents the span of an observation window (here shown to include  $5 \times 5$  LR pixel spacings). The small box represents the estimation window containing a set of  $6 \times 6$  HR pixels to estimate based on the LR pixels in the observation window.

a weighted sum of the LR pixels in the observation window. This is expressed as

$$\hat{\mathbf{d}}_i = \mathbf{W}_i^T \mathbf{g}_i, \quad (6)$$

where  $\hat{\mathbf{d}}_i = [\hat{d}_{i,1}, \hat{d}_{i,2}, \dots, \hat{d}_{i,D_x D_y}]^T$  and  $\mathbf{W}_i$  is a  $K_i \times D_x D_y$  matrix of weights. Each column of  $\mathbf{W}_i$  contains the weights used for a particular HR pixel within the estimation window. If we define the  $j$ 'th column of  $\mathbf{W}_i$  to be  $\mathbf{w}_{i,j}$ , then  $\hat{d}_{i,j} = \mathbf{w}_{i,j}^T \mathbf{g}_i$ . Within this framework, a potentially unique set of weights can be used to form each HR pixel estimate. This allows the filter to adapt to any changing intensity statistics and varying spatial distributions of the LR pixels on the HR grid. The observation window moves across the HR grid in steps of  $D_x$  and  $D_y$  in the horizontal and vertical directions, respectively. As will be shown in Section IV, there are computational savings associated with estimating more than one HR pixel per observation window (i.e., letting  $D_x, D_y > 1$ ). All of these outputs are combined to form an estimate of the ideally sampled image  $\mathbf{z}$ , and this estimate is denoted  $\hat{\mathbf{z}}$  as shown in Fig. 3.

The weights in (6) that minimize the mean squared error are found in a fashion similar to that for

an FIR Wiener filter. However, unlike a standard Wiener filter, these weights vary based on the spatial distribution of LR pixels within the observation window. It can be shown [14, 17, 18] that the weights that minimize the mean squared error are given by

$$\mathbf{W}_i = \mathbf{R}_i^{-1} \mathbf{P}_i, \quad (7)$$

where  $\mathbf{R}_i = E\{\mathbf{g}_i \mathbf{g}_i^T\}$  is the autocorrelation matrix for the observation vector and  $\mathbf{P}_i = E\{\mathbf{g}_i \mathbf{d}_i^T\}$  is the cross-correlation between the desired vector  $\mathbf{d}_i = [d_{i,1}, d_{i,2}, \dots, d_{i,D_x D_y}]^T$  and the observation vector. Note that  $\mathbf{d}_i$  is the vector of desired HR samples corresponding to the HR grid positions in the estimation window. Note that before applying the weights in (7), each column in  $\mathbf{W}_i$  is normalized so that it sums to 1 to ensure continuity across observation windows and to avoid potential grid pattern artifacts. In [17], a similar weight matrix was determined using empirical estimates from training images where the LR pixels are forced to lie exactly on HR pixels spacings on the HR grid. Here we use a parametric continuous model for the desired image 2d autocorrelation function and cross-correlation function. We then sample these continuous functions at coordinates determined by the locations of the LR pixels on the HR grid to fill  $\mathbf{R}_i$  and  $\mathbf{P}_i$ . This eliminates the necessity for training images and gives the system a relatively small number of convenient tuning parameters (parameters that define the correlation functions). These tuning parameters provide a level of control over the characteristics of the HR image estimate.

### B. Global Statistical Model

To begin the development of modeling  $\mathbf{R}_i$  and  $\mathbf{P}_i$ , first let  $\mathbf{f}_i$ , be the noise-free version of the  $i$ 'th observation vector  $\mathbf{g}_i$ . In other words, we have  $\mathbf{g}_i = \mathbf{f}_i + \mathbf{n}_i$ , where  $\mathbf{n}_i$  is the random noise vector associated with the samples within observation window  $i$ . Furthermore, assume that the noise is a zero-mean uncorrelated noise vector with independent and identically distributed elements with variance  $\sigma_n^2$ . In this case, it is straightforward to show that the autocorrelation matrix for the observation vector is given by

$$\mathbf{R}_i = E\{\mathbf{g}_i \mathbf{g}_i^T\} = E\{\mathbf{f}_i \mathbf{f}_i^T\} + \sigma_n^2 \mathbf{I} \quad (8)$$

and cross-correlation matrix is given by

$$\mathbf{P}_i = E\{\mathbf{g}_i \mathbf{d}_i^T\} = E\{\mathbf{f}_i \mathbf{d}_i^T\}. \quad (9)$$

Now the problem reduces to modeling the elements of  $E\{\mathbf{f}_i \mathbf{f}_i^T\}$  and  $E\{\mathbf{f}_i \mathbf{d}_i^T\}$ . To do so, consider a wide sense stationary autocorrelation function,  $r_{dd}(x, y)$ , for the desired image  $d(x, y)$ . The cross-correlation

function between  $d(x, y)$  and  $f(x, y)$ , as shown in Fig. 2, can be expressed in terms of  $r_{dd}(x, y)$  [27] as

$$r_{df}(x, y) = r_{dd}(x, y) * h(x, y). \quad (10)$$

The autocorrelation of  $f(x, y)$  is given by

$$r_{ff}(x, y) = r_{dd}(x, y) * h(x, y) * h(-x, -y). \quad (11)$$

With knowledge of the positions of the LR pixels that comprise  $\mathbf{g}_i$  (and  $\mathbf{f}_i$ ), the horizontal and vertical distances between all the samples can easily be computed. Evaluating (11) using all these displacements yields  $E\{\mathbf{f}_i \mathbf{f}_i^T\}$ , and then in turn  $\mathbf{R}_i$  can be found from (8). With knowledge of the horizontal and vertical distances between the LR pixels in the observation window and the HR pixels in the estimation window, one can evaluate (10) at the appropriate coordinates to fill  $\mathbf{P}_i$  as defined in (9). Note that as the relative positions of the LR pixels in the observation window change across an HR grid or between HR frames, so do the autocorrelation and cross-correlation matrices. This leads to weights tuned to the specific spatial pattern of LR pixels.

The proposed framework allows for the selection of the desired image autocorrelations,  $r_{dd}(x, y)$ . This can be obtained empirically from statistically representative training images or defined using a parametric model. For the experimental results presented in Section V, we employ a circularly-symmetric parametric autocorrelation model [28] of the form

$$r_{dd}(x, y) = \sigma_d^2 \rho^{\sqrt{x^2 + y^2}}, \quad (12)$$

where  $\sigma_d^2$  is the variance of the desired image and  $\rho$  is a tuning parameter that controls the decay of the autocorrelation with distance. We have found that this model is representative of empirically estimated image autocorrelation functions for a wide range of natural images (including the images used here). A plot of this autocorrelation function is shown in Fig. 5 for  $\sigma_d^2 = 1$  and  $\rho = 0.75$ . Thus, given the model parameters,  $\sigma_d^2$  and  $\rho$ , we can use (12) to get (10) and (11). Sampling these according the spatial distribution of LR pixels in any given observation window gives rise to  $\mathbf{R}_i$  and  $\mathbf{P}_i$ . Using these matrices in (7) generates the weights for a given observation vector. Finally, the HR pixel estimates are formed using (6). This process can be repeated for each position of the observation window on the HR grid. However, for translational motion, a repeating pattern of spatial locations for the LR pixels occurs, as can be seen in Fig. 4. The number of unique patterns ranges from 1 to  $L_x L_y$ , depending on the choice of  $D_x$  and  $D_y$ , as will be addressed in Section IV. Thus, a weight matrix need only be obtained once for each unique spatial pattern (and not necessarily once per observation vector). This greatly reduces the computational complexity of the algorithm as described in Section IV.

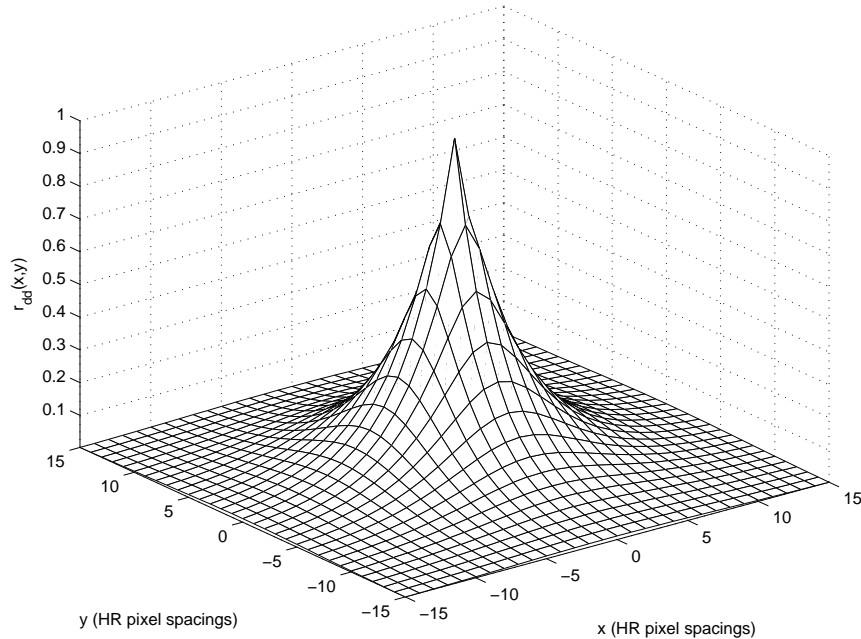


Fig. 5. Autocorrelation function model  $r_{dd}(x, y)$  for  $\sigma_d^2 = 1$  and  $\rho = 0.75$

### C. Spatially Varying Statistical Model

While we believe the method described above is very useful, it is based on the assumption that the random processes involved are wide sense stationary. Since many images do not obey this assumption, it may be advantageous in some cases to employ a desired autocorrelation model that is a function of the observation window position. Thus, in addition to adapting to the local spatial distribution of the LR pixels in each observation window, the weights would also adapt to the local intensity statistics of those samples. Here we fix  $\rho$  in (12) and allow the local desired image variance parameter to vary with the position of the observation window. The resulting autocorrelation model can be expressed as

$$r_{dd_i}(x, y) = \sigma_{d_i}^2 \rho^{\sqrt{x^2 + y^2}}, \quad (13)$$

where  $\sigma_{d_i}^2$  is the variance of the local region of the desired image that is responsible for producing  $\mathbf{g}_i$ . To make use of this spatially varying model, we must estimate the model parameter  $\sigma_{d_i}^2$  for each observation vector.

Using (11), it can be shown that the relationship between the desired image variance,  $\sigma_{d_i}^2$ , and the

variance of the corresponding PSF degraded image,  $\sigma_{f_i}^2$ , is given by

$$\sigma_{d_i}^2 = \frac{1}{C(\rho)} \sigma_{f_i}^2, \quad (14)$$

where

$$C(\rho) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \rho^{\sqrt{x^2+y^2}} \tilde{h}(x, y) dx dy, \quad (15)$$

and  $\tilde{h}(x, y) = h(x, y) * h(-x, -y)$ . Thus, if we can estimate  $\sigma_{f_i}^2$ , we can then use (14) to form a corresponding estimate of  $\sigma_{d_i}^2$ . Since the noise is assumed to be independent, we estimate  $\sigma_{f_i}^2$  as

$$\hat{\sigma}_{f_i}^2 = \hat{\sigma}_{g_i}^2 - \sigma_n^2, \quad (16)$$

where  $\hat{\sigma}_{g_i}^2$  is the sample variance estimate from the elements of the observation vector  $\mathbf{g}_i$ .

A plot showing the empirical relationship between  $\sigma_{f_i}$  and  $\sigma_{d_i}$  is provided in Fig. 6. These sample standard deviations are obtained from local windows of images derived from the 8 bit aerial image in Fig. 9(a). For these data  $L_x = L_y = 4$ ,  $W_x = W_y = 12$ , and  $h(x, y) = \frac{1}{16} \text{rect}(\frac{x}{4}, \frac{y}{4})$ , where  $x$  and  $y$  are measured in HR pixel spacings. Also shown in Fig. 6 is the linear mapping defined by (14) for  $\rho = 0.75$  and a cubic polynomial curve fit of the data. Note that the linear model from (14) is fairly accurate when applied to these data, but the cubic fit is better. Thus, if statistically representative training data are available, it may be advantageous to use such an empirically based polynomial mapping rather than (14) to estimate  $\sigma_{d_i}^2$ . This suggests that the model in (13) may not be able to fully account for the non-stationary characteristics of this image using a constant  $\rho$ . Attempting to estimate a spatially varying  $\rho$  and  $\sigma_d^2$  is one area of ongoing investigation. However, we believe that the model with a fixed  $\rho$  is sufficiently accurate to yield useful results, especially when an polynomial mapping function is used to estimate  $\sigma_{d_i}^2$  from (16).

When a different set of weights is already being computed for each observation vector, including the spatially varying statistical model adds little computational complexity. However, for translational motion, we wish to exploit the repeating spatial pattern of LR pixels and not compute a different set of weights for each observation window. One option is to quantize  $\sigma_{d_i}^2$  to  $Q$  discrete levels. This would require that only  $Q$  weight matrices be computed for each unique spatial pattern. This provides a practical way to employ the spatially varying statistical model without the need to compute distinct weights for each observation window.

Note that if  $E\{\mathbf{f}_i \mathbf{f}_i^T\}$  and  $E\{\mathbf{f}_i \mathbf{d}_i^T\}$  are obtained using the model in (12) with  $\sigma_d^2 = 1$ , one can conveniently obtain  $\mathbf{R}_i$  and  $\mathbf{P}_i$  for any  $\sigma_{d_i}^2$  as

$$\mathbf{R}_i = E\{\mathbf{g}_i \mathbf{g}_i^T\} = \sigma_{d_i}^2 E\{\mathbf{f}_i \mathbf{f}_i^T\} + \sigma_n^2 \mathbf{I} \quad (17)$$

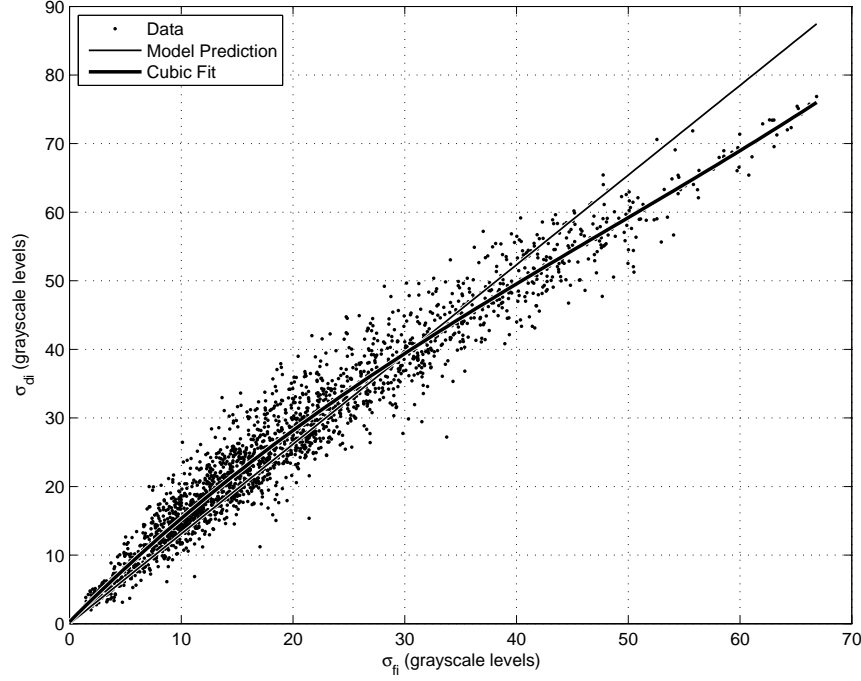


Fig. 6. Scatter plot showing the empirical relationship between  $\sigma_{f_i}$  and  $\sigma_{d_i}$  based on the 8 bit aerial image in Fig. 9(b). For these data,  $L_x = L_y = 4$ ,  $W_x = W_y = 12$ , and  $h(x, y) = \frac{1}{16}\text{rect}(\frac{x}{4}, \frac{y}{4})$ , where  $x$  and  $y$  are measured in HR pixel spacings.

and

$$\mathbf{P}_i = E\{\mathbf{g}_i \mathbf{d}_i^T\} = \sigma_{d_i}^2 E\{\mathbf{f}_i \mathbf{d}_i^T\}, \quad (18)$$

respectively. The filter weights are given by

$$\mathbf{W}_i = [E\{\mathbf{f}_i \mathbf{f}_i^T\} + \frac{\sigma_n^2}{\sigma_{d_i}^2} \mathbf{I}]^{-1} E\{\mathbf{f}_i \mathbf{d}_i^T\}. \quad (19)$$

The coefficient of the identity matrix is a noise-to-signal (NSR) ratio. Thus, letting  $\sigma_{d_i}^2$  vary between observation vectors allows the filter weights to adapt to a varying NSR across the HR grid. To help illustrate the impact of the NSR on the filter weights, Fig. 7 shows a stem plot of the weights for the case where  $P = 3$ ,  $L_x = L_y = 3$ ,  $W_x = W_y = 9$ , and  $h(x, y) = \frac{1}{9}\text{rect}(\frac{x}{3}, \frac{y}{3})$ , where  $x$  and  $y$  are measured in HR pixel spacings. In these plots, a stem is located at the position of each LR pixel on the HR grid and the height of each stem is the corresponding weight. Common symbols correspond to LR pixels from a common frame and the “x” symbol at (0,0) marks the location HR pixel being estimated. Figures 7(a) and (b) show the weights for NSR = 1.0 and NSR = 0.1, respectively. Note that for very noisy data (e.g., NSR = 1), the weights are more evenly distributed, producing a smoothing effect. In

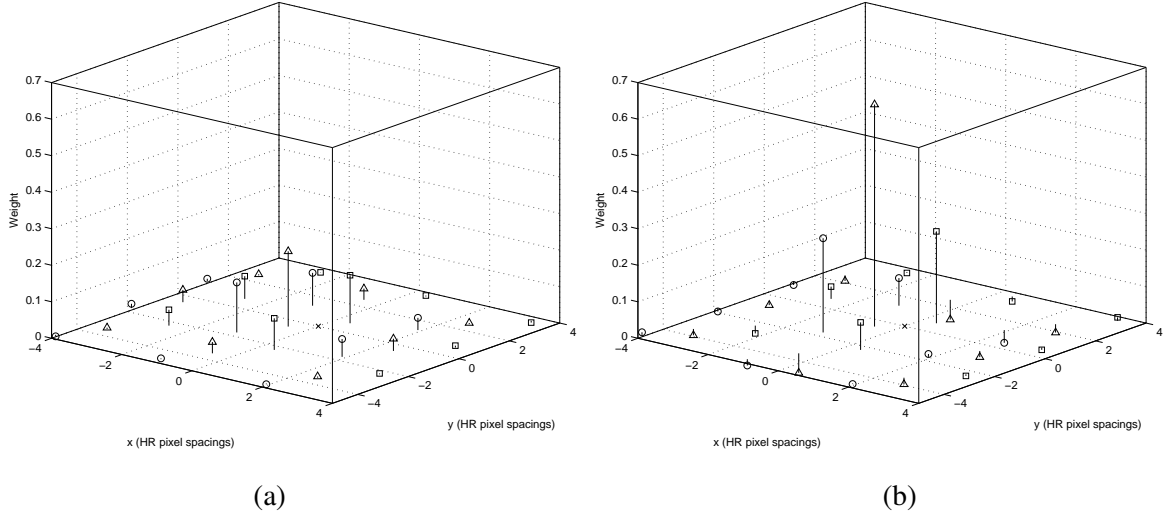


Fig. 7. Weights for the case where  $P = 3$ ,  $L_x = L_y = 3$ ,  $W_x = W_y = 9$ , and  $h(x, y) = \frac{1}{9}\text{rect}(\frac{x}{3}, \frac{y}{3})$ , where  $x$  and  $y$  are measured in HR pixel spacings. (a) NSR = 1.0 (b) NSR = 0.1

contrast, the NSR = 0.1 plot shows much more weight given to the LR samples in close proximity to the HR pixel being estimated. It is also interesting to note that if the spatial location of any of the LR pixels were to change, the weights for all the LR pixels change in an effort to make the best use of the available samples in an MSE sense. Thus, these weights are not simply samples of a fixed underlying interpolation kernel.

#### IV. COMPUTATIONAL COMPLEXITY

In this section, we perform a computational complexity analysis of AWF SR algorithm (excluding the registration step which is common to most SR methods). There are two main components of the system, one is computing the appropriate filter weights and the other is applying those weights to form the output. The analysis here is based on the number floating point multiply and add operations (flops) required to generate an HR output frame.

Let us first consider the case of translational motion where the observation window dimensions,  $W_x$  and  $W_y$ , are a multiple of  $L_x$  and  $L_y$ , respectively. In this case, the number of flops associated with applying the weights to get an output HR frame is  $NK$ , where  $K$  is given by (5) and  $N$  is the total number of HR pixels to be estimated. We must also consider the computation of the filter weights. With only translation motion between the LR frames, the HR grid exhibits a repetitive structure in the region where the LR frames overlap. This means that only a limited number of distinct weight matrices need to be



computed. When the estimation window size is not a multiple of the upsampling factor (i.e.,  $D_x \neq mL_x$  and  $D_y \neq mL_y$ , where  $n, m$  are positive integers), there exist  $L_x L_y$  unique LR pixel distributions for the observation windows moving across the HR grid. When  $D_x = mL_x$  and  $D_y = nL_y$ , as shown in Fig. 4, there is only one spatial distribution of LR pixels encountered as the observation window moves across the HR grid by steps of  $D_x$  horizontally and  $D_y$  vertically.

Computing the weights for each unique LR pixel distribution requires filling the matrices  $\mathbf{R}_i$  and  $\mathbf{P}_i$  and solving the equations  $\mathbf{R}_i \mathbf{W}_i = \mathbf{P}_i$ . Filling  $\mathbf{R}_i$  and  $\mathbf{P}_i$  can be done using a look-up-table with interpolation. These operations are neglected in the following computational complexity analysis. The weights can be computed using Cholesky factorization which requires  $K^3/3$  flops to perform LU decomposition for the  $K \times K$  autocorrelation matrix and  $2K^2$  flops to solve for one set of weights using forward and backward substitution. Thus, considering only translational motion the total flop count is given by

$$\text{flops} \cong QL_x L_y \left( \frac{K^3}{3} + 2D_x D_y K^2 \right) + NK, \quad (20)$$

when  $D_x$  and  $D_y$  are not integer multiples of  $L_x$  and  $L_y$ , respectively. When  $D_x$  and  $D_y$  are integer multiples of  $L_x$  and  $L_y$ , respectively, the flop count is given by

$$\text{flops} \cong \frac{QK^3}{3} + 2QD_x D_y K^2 + NK. \quad (21)$$

In comparison, the weighted nearest neighbors (WNN) algorithm [3,4] requires  $lN$  flops for the nonuniform interpolation, where  $l$  is the number of neighbors used. It then requires a 2d FFT,  $N$  multiplies, and an inverse 2d FFT. Using  $10N \log_2(\sqrt{N})$  as the approximate flop count for the 2d FFT, the total flop count for the WNN algorithm is given by

$$\text{flops} \cong \left( 20 \log_2(\sqrt{N}) + l + 1 \right) N. \quad (22)$$

Figure 8 shows the flop counts as a function of the HR image dimension  $\sqrt{N}$  for both the proposed AWF SR and the WNN algorithm using  $l = 4$  neighbors. For all of the AWF SR curves  $L_x = L_y = D_x = D_y = 4$  and  $Q = 1$ . Consider the AW SR algorithm with parameters  $P = 16$  and  $W_x = W_y = 12$ . In this case the AWF SR and WNN algorithms have a comparable complexity for  $\sqrt{N} = 300$ . Note that increasing  $\sqrt{N}$  favors the AWF SR algorithm in this case. Since the complexity of the WNN algorithm is dominated by the FFT, there is little one can do to control the computational complexity. However, as can be seen with the AWF SR method, there are a number of parameters that can be adjusted to change the computational complexity. The main complexity controlling parameters are  $P$ ,  $Q$ , and the window dimensions,  $W_x$  and  $W_y$ .

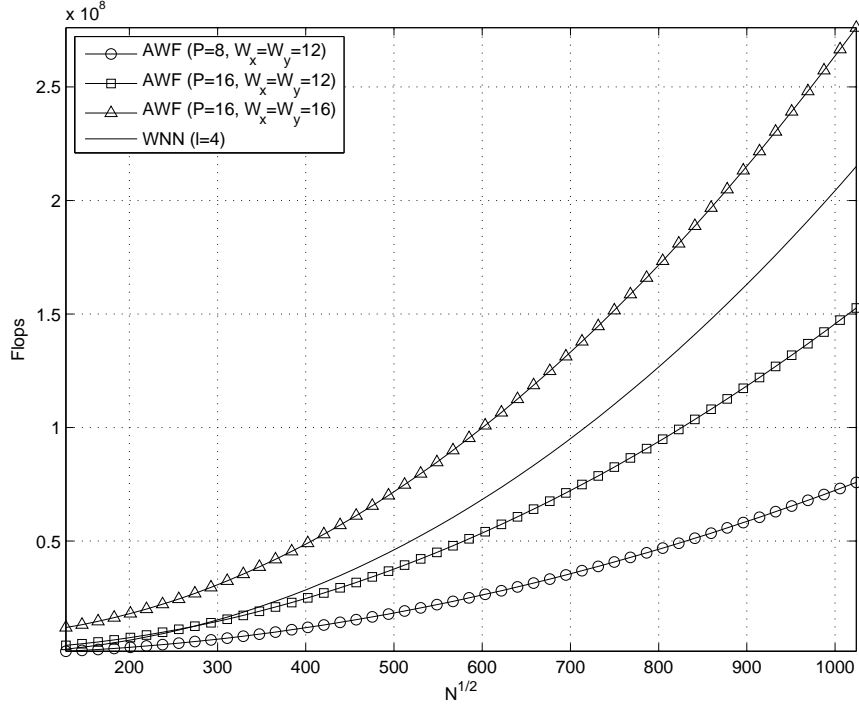


Fig. 8. Flop counts as a function of the HR image dimension  $\sqrt{N}$  for both the proposed AWF SR and the WNN algorithm. For the AWF SR curves,  $L_x = L_y = D_x = D_y = 4$  and  $Q = 1$ . For the WNN,  $l = 4$ .

When the motion between the LR frames is other than global translational motion, the regularity in the HR grid pattern may not be preserved and the number of LR pixels in the observation window may vary. In the worst case scenario, a weight matrix must be computed for every estimation window in the HR grid. In particular, the LU factorization of the autocorrelation matrix is performed for every estimation window in the HR grid and the forward and backward substitution is carried out for every HR pixel. In this case, the total number of flops per output pixel required is approximately given by

$$\text{flops} \approx \frac{N\bar{K}^3}{3D_xD_y} + 2N\bar{K}^2 + N\bar{K}, \quad (23)$$

where  $\bar{K}$  is the average number of LR pixels within an observation window. Clearly, the number of flops is now dominated by the Cholesky factorization and is significantly higher than that for translational motion. However, each observation window can be processed independently in parallel. Thus, this algorithm may be suitable for real-time or near real-time implementation using a massively parallel hardware architecture.

It should be noted that computational complexity of the PWS filter implementation in [17] with  $M = 1$  is essentially the same computational complexity as the AWF with  $Q = 1$ . However, when the PWS employs multiple partitions, that method has the added burden of performing vector quantization on each

of the observation vectors. The vector quantization processing used in [17] also increases the complexity of filter training by requiring the design of a vector quantization codebook. The autocorrelation and cross-correlation statistics must also be estimated empirically from training images prior to filtering using the method in [17]. The amount of training data required grows with the number of partitions (since data for each partition are required). In contrast, the proposed AWF SR method requires only an estimate of the local variance for each observation vector in order to obtain spatially varying statistics and requires only  $\rho$  and  $\sigma_n^2$  to be ready to process a set of frames. Also,  $Q$  can be increased as desired without the need for training data or fear of poorly conditioned autocorrelation matrices.

## V. EXPERIMENTAL RESULTS

In this section, we demonstrate the efficacy of the proposed AWF SR algorithm using both simulated image sequences and a real video sequence obtained with a forward looking infrared (FLIR) imager. The simulated sequences are derived from a still frame aerial image and allow for quantitative error analysis. The FLIR sequence allows us to illustrate the performance of the proposed SR algorithm in a real application. To help put the results in context, we compare the performance of the AWF SR algorithm to several previously published techniques. The benchmark methods consist of the sample-and-add method with Wiener deconvolution (SA+Wiener) [12], the WNN method [3, 4], the SKWF method [16], and the PWS filter approach [17]. The SA+Wiener is implemented using the software in [29], and the SKWF implementation is the author's interpretation of one method described in [16]. These methods were selected because they have a relatively low computational complexity, approximately on par with that of the proposed method. We also include some results with the regularized least squares (RLS) image reconstruction algorithm [1]. This serves as a representative of the more computationally demanding iterative SR methods. Single-frame bicubic interpolation is used to provide a performance baseline for evaluating all of the multi-frame SR methods.

### A. Simulated Data

To generate the simulated video sequences, we use the 8 bit image shown in Fig. 9(a) and follow the steps in the observation model shown in Fig. 1. In particular, the HR image in Fig. 9(a) is translationally shifted by subpixel spacings using bicubic interpolation, blurred with a moving average filter of size  $L_x \times L_y$ , and then down-sampled by  $L_x$  and  $L_y$  in the horizontal and vertical directions, respectively. The moving average filter models the detector integration for square detectors (and neglects diffraction). This is a fairly good approximation for heavily undersampled systems, where detectors are much larger

than the diffraction-limited spot size from the optics [1]. A simulated LR frame with no noise is shown in Fig. 9(c) for  $L_x = L_y = 4$ . The same frame with additive noise of variance  $\sigma_n^2 = 100$  is shown in Fig. 9(d). A set of 16 frames are obtained in this manner using the random translational shifts shown in Fig. 10.

The first experiment compares the various SR methods using different numbers of input frames. The MSE and MAE as a function of  $P$  are shown in Figs. 11 and Fig. 12, respectively. The noise variance for these results is  $\sigma_n^2 = 100$ . For each of the methods, there are one or more tuning parameters related to NSR that affect system performance. For these methods, the noise related parameters must account for all noise sources, including that effectively produced by registration errors. Thus, using the additive Gaussian noise variance (if known) may not lead to the lowest errors. To provide as fair a comparison as possible, we have attempted to optimize these parameters to get the best performance from each algorithm (in an MSE sense) for each  $P$ . Other statistical parameters, such as the polynomial fitting parameters for estimating  $\sigma_{d_i}^2$  for the AWF and the empirical autocorrelation and cross-correlation for the PWS, are estimated using the independent training image shown in Fig. 9(b). Note that we use  $\rho = 0.75$  for the AWF method and we employ a window size of  $W_x = W_y = 12$  and an estimation window of size  $D_x = D_y = 4$ . We employ the same size observation window for the PWS and SKWF methods and the PWS uses an  $8 \times 8$  partitioning window.

The results in Figs. 11 and 12 show that all the methods improve with an increased number of input frames and the errors are much lower than for single frame interpolation. The AWF with ( $Q = 20$ ) consistently provides the lowest MAE and MSE. The PWS with 10 partitions ( $M = 10$ ) produces errors in this study that are higher than those for the new AWF ( $Q = 20$ ) method. We believe this can be partly attributed to the quantization of the LR frame positions to a discrete HR grid used by the PWS. This positional quantization is essentially introducing another noise source that hurts performance. Also, the vector quantization used by the PWS does not perform as well with fewer frames or when the HR grid is sparsely populated. The increased error for the the WNN and SA+Wiener methods may be partly attributable to how these methods perform the nonuniform interpolation and restoration in two independent steps. For the WNN, the interpolation is based only on the distance of nearest LR pixels using a simple inverse distance weighting. The SKWF uses a spatially invariant filter on the HR grid and does not independently optimize the weights for each HR pixel.

The output images corresponding to the  $P = 16$  point in Figs. 11 and 12 are shown in Fig. 13. The output obtained with single frame bicubic interpolation is shown in Fig. 13(a). The output of the SA+Wiener method is shown in Fig. 13(b) for an NSR of 0.17. The WNN output with an NSR of 0.14

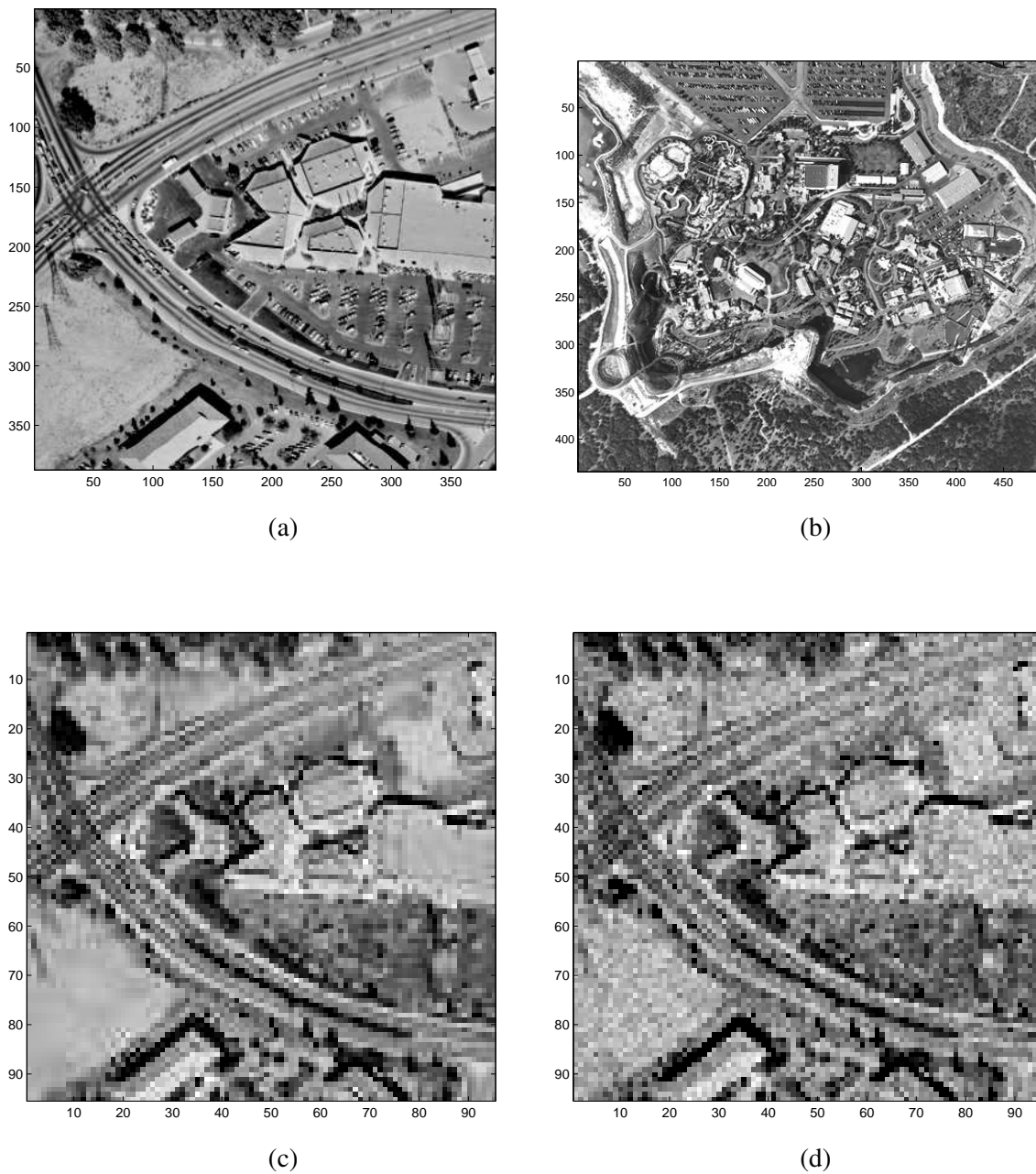


Fig. 9. Images used for error analysis with simulated video sequences. (a) Test image used to create an image sequence with translational motion, (b) training image used to estimate statistical parameters, (c) simulated LR frame with  $L_x = L_y = 4$  and no noise, and (d) simulated LR frame with  $L_x = L_y = 4$  and noise with variance  $\sigma_n^2 = 100$ .

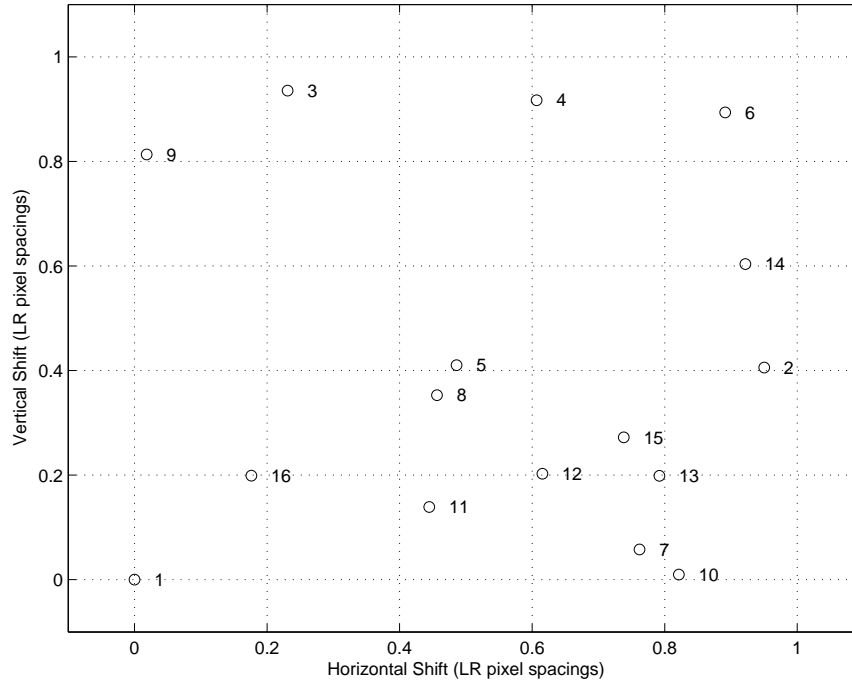


Fig. 10. Translational shifts for 16 simulated LR frames.

is shown in Fig. 13(c). The SKWF output is shown in Fig. 13(d) using  $\rho = 1$  (as defined in [16]) and noise power spectral density value of 0.048. The PWS with  $M = 10$  and modeled noise variance of 97.37 is shown in Fig. 13(e). Finally, the AWF SR output with  $Q = 20$  and modeled noise variance of 73.68 (as defined in (19)), is shown in Fig. 13(f). It is interesting to note that the image in Fig. 13(f) appears smoother than the others in the flat regions, such as on the roof tops, parking lots, and the field in the lower left. This is due to the fact that these areas have lower variance and the AWF with  $Q = 20$  does more smoothing there and less smoothing in the dynamic areas with higher variance. Thus, the use of spatially varying statistics does appear to impact the subjective quality of the images, at least when a moderate to high level of noise is present. When applying the AWF with  $Q = 1$  to this same dataset, the optimum global NSR is 0.084 compared with an average NSR of 0.079 for  $Q = 20$ . Note also that 33.6% of the local windows for the  $Q = 20$  adaptive AWF use weights computed with an NSR in (19) lower than that for the global AWF (some dramatically so). Weights computed for the AWF with lower NSRs will tend to do more sharpening and less smoothing. Thus, the adaptive AWF is doing more than just extra smoothing, it is also doing extra sharpening in 33.6% of the local windows.

In order to evaluate how sensitive the SR methods are to LR frame shift pattern variations, we have

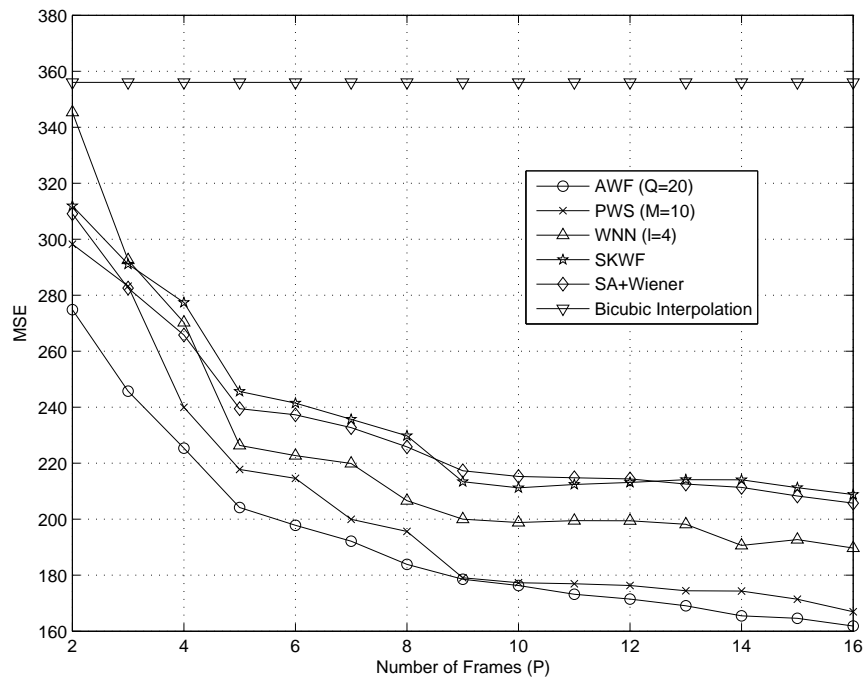


Fig. 11. Mean squared error as a function of the number of LR frames used for simulated image sequence with additive noise of variance  $\sigma_n^2 = 100$ .

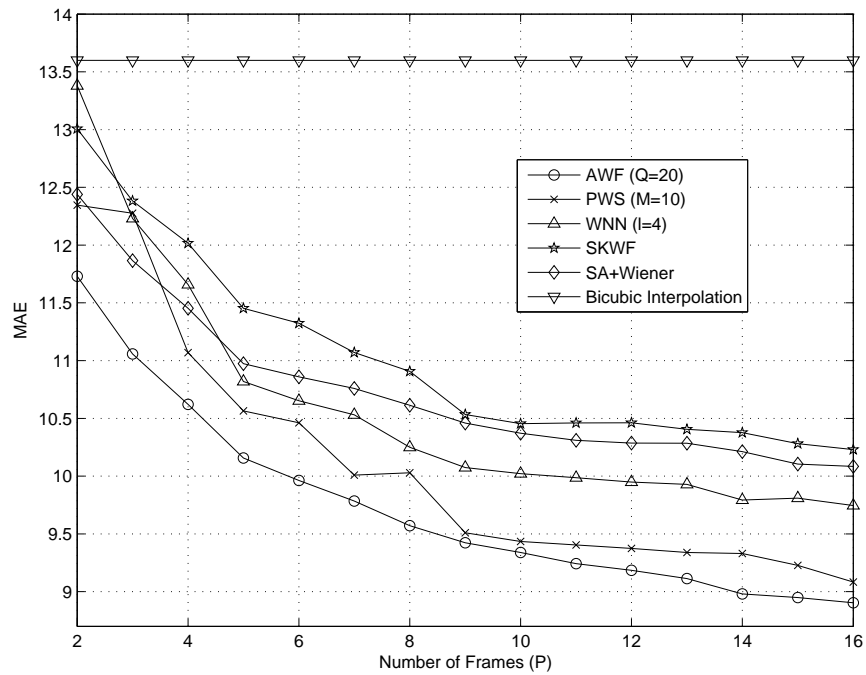


Fig. 12. Mean absolute error as a function of the number of LR frames used for simulated image sequence with additive noise of variance  $\sigma_n^2 = 100$ .

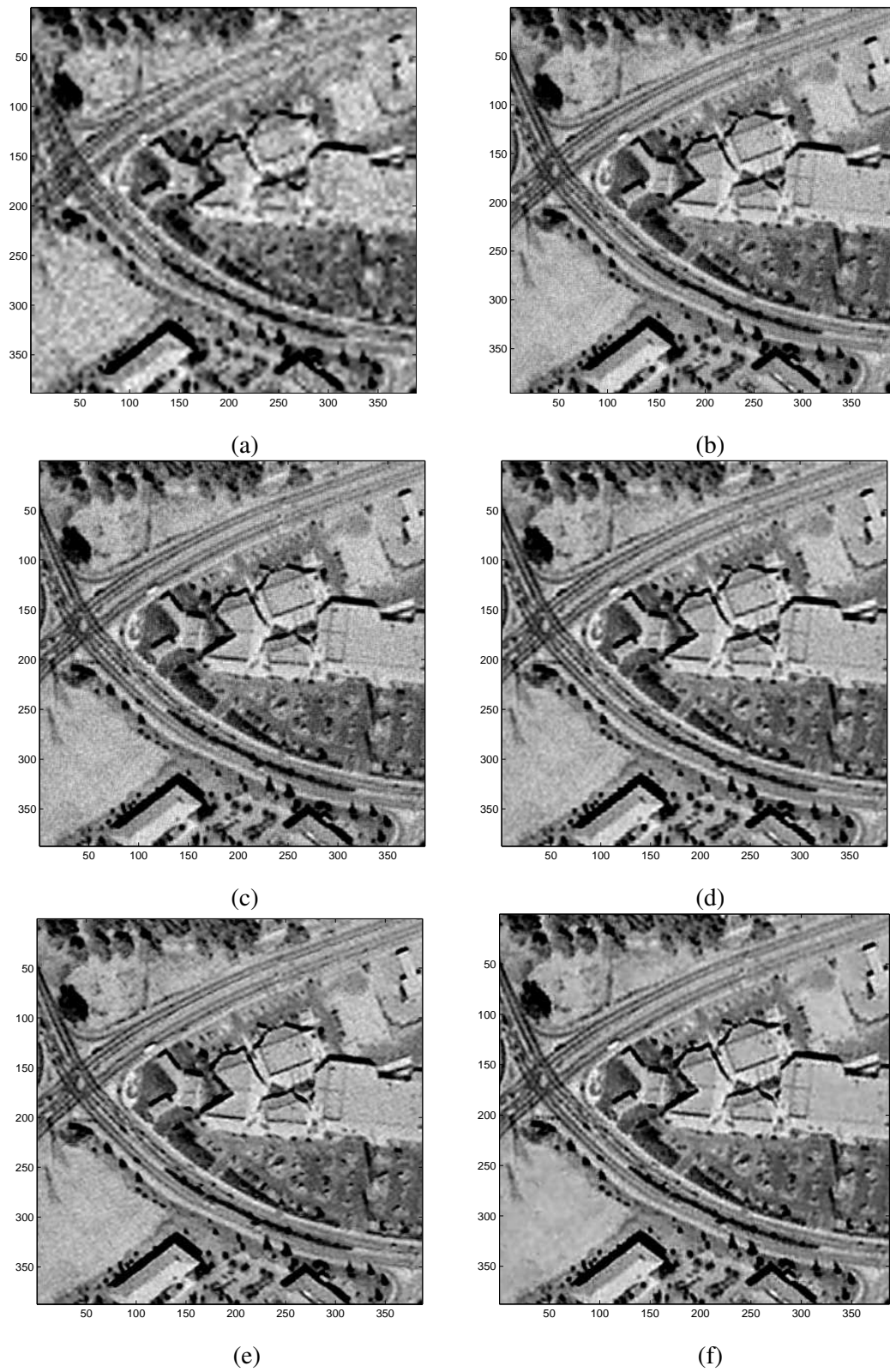


Fig. 13. Output images for the simulated image sequence with  $P = 16$ . (a) Single frame bicubic interpolation, (b) SA + Wiener, (c) WNN with  $l = 4$ , (d) SKWF, (e) PWS with  $M = 10$ , and (f) AWF with  $Q = 20$ .



TABLE I

MSE AND MAE STATISTICS FOR THE VARIOUS SR ALGORITHMS OVER 100 SHIFT PATTERN REALIZATIONS.

Algorithm	MSE		MAE	
	Mean	Std	Mean	Std
AWF (Q=1)	163.19	2.48	9.17	0.06
AWF (Q=20)	158.00	2.41	8.82	0.06
PWS (M=1)	172.97	4.83	9.43	0.12
PWS (M=10)	164.20	6.71	9.06	0.23
WNN (l=4)	183.32	5.36	9.68	0.13
SKWF	203.61	14.52	10.15	0.30
RLS	161.01	2.61	9.24	0.06
Bicubic	360.36	0.00	13.66	0.00

done a quantitative analysis using 100 independent sets of LR input frames. Each input set consists of 16 frames obtained using the same observation model as that for the results above. The spatial pattern of the LR frames varies randomly between input sets. This is important for video applications where different groups of frames will have different motion parameters and we are interested in how robust the SR methods are to the varying shift pattern. The parameters used for the results in Fig. 13 are applied to all 100 input sets. The sample mean and standard deviation of the error metrics are shown in Table I for the SR methods. Here we include the results for the PWS with  $M = 1$  using a modeled noise variance of 152.63 and the AWF with  $Q = 1$  (global statistics) with a modeled noise variance of 105.26. We also include results for the RLS algorithm for reference using 10 iterations with  $\lambda = 0.49$  [1]. Table I shows that the AWF with  $Q = 20$  not only produces the lowest MSE and MAE, but also the lowest error standard deviations. The PWS with  $M = 10$  produces lower errors than the PWS with  $M = 1$ , but note that the error standard deviation goes up and is 2.8 times that of the AWF for  $Q = 20$ . The higher error and error variance for the PWS compared with the AWF may be partly attributable to the quantization of motion parameters used by the PWS. For some input patterns, there may be little quantization effects if the shifts happen to be near integer multiples of HR pixel spacings. In other cases the quantization effect can be more pronounced. This effect tends to increase the standard deviation of the error metrics for this approach. The results in Table I may also suggest that the vector quantization based partitioning is more sensitive to the shift pattern than the local NSR approach of the AWF.

### B. FLIR Video Data

Here we apply the various SR algorithms to video frames from a FLIR imager. These frames have translational motion introduced by panning across a scene. These data have been provided courtesy of the Sensors Directorate at the Air Force Research Laboratories, Wright Patterson Air Force Base. The FLIR imager is equipped with a  $128 \times 128$  Amber AE-4128 infrared FPA with Indium-Antimonide (InSb) detectors and  $f/3$  optics. For a  $4 \mu\text{m}$  wavelength, the optical cut-off frequency is 83.3 cycles/mm. The Amber FPA has square detectors with a pitch of 0.050 mm (with 80% fill factor), yielding a sampling frequency of 20 cycles/mm.

A group of  $P = 16$  frames is used and each LR frame has been cropped to a size of  $64 \times 64$ . Figure 14 shows one LR frame from the sequence. The estimated translational shift parameters are shown in Fig. 15. We process the 16 frames to produce an HR image with  $L_x = L_y = 5$ . The AWF, PWS, and SKWF all employ a  $15 \times 15$  observation window. The AWF and PWS use a  $5 \times 5$  estimation window, and the PWS uses a  $7 \times 7$  partitioning window. The system PSF used is based on a diffraction-limited model for the optics and the shape of the active area of the detectors, as described in [1]. The parameters for all the SR methods are based on optimization with the simulated imagery with a noise variance of  $\sigma_n^2 = 10$ . The outputs for the AWF and the benchmark methods are shown in Fig. 16. The NSRs for the SA+Wiener and WNN are 0.080 and 0.058, respectively. The SKWF output is using  $\rho = 1$  (as defined in [16]) and noise power spectral density value of 0.043. The PWS result is for  $M = 10$  and a modeled noise variance of 16.32 and the AWF results is for  $Q = 20$  and modeled noise variance of 10.0. In the single frame interpolation shown in Fig. 16(a), note the severe ‘‘checkerboard’’ aliasing artifacts on the roof of the main building. These artifacts are virtually eliminated with all of the multi-frame SR methods. The AWF image in Fig. 16(f) appears to have fewer artifacts than the other images.

## VI. CONCLUSIONS

We have proposed a novel SR algorithm based on a type of adaptive Wiener filter. Multiple frames are acquired and registered to a common continuous HR grid (without quantizing the motion parameters). The estimated HR pixels are formed as a weighted sum of neighboring LR pixels in an observation window than scans across the HR grid. For each HR pixel estimate, the weights are optimized for the spatial pattern of LR pixels in each observation window relative to the HR pixel. Here we consider both a global statistical model and a spatially varying model. When using a spatially varying statistical model, the proposed approach not only adapts to the local spatial pattern of LR pixels, but also to the local

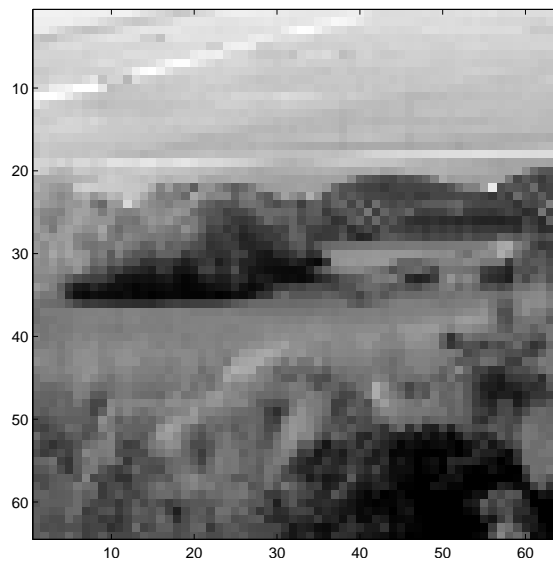


Fig. 14. Single  $64 \times 64$  frame in the FLIR image sequence.

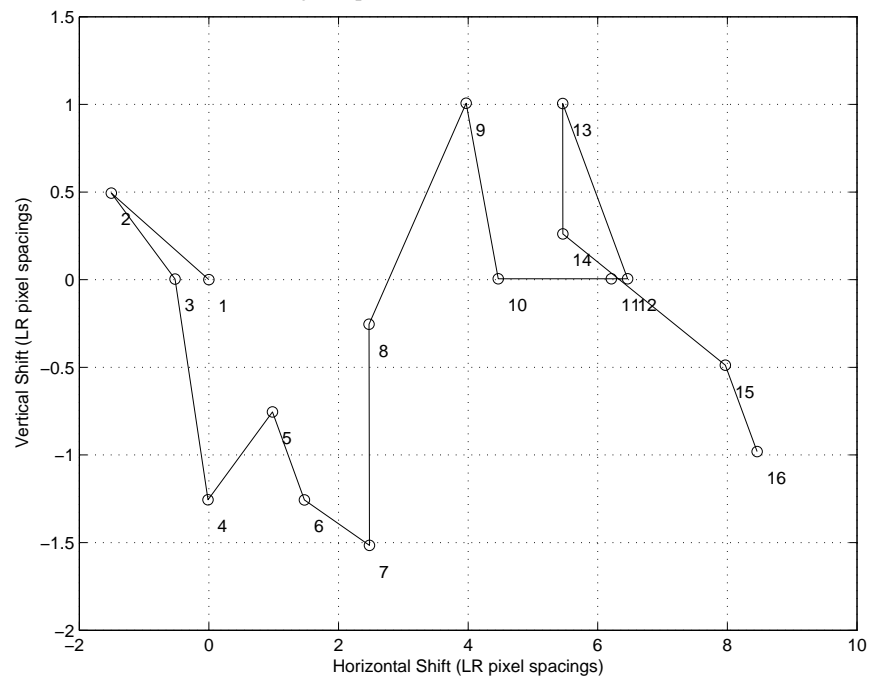


Fig. 15. Estimated translational shifts (in LR pixel spacings) for the FLIR image sequence.

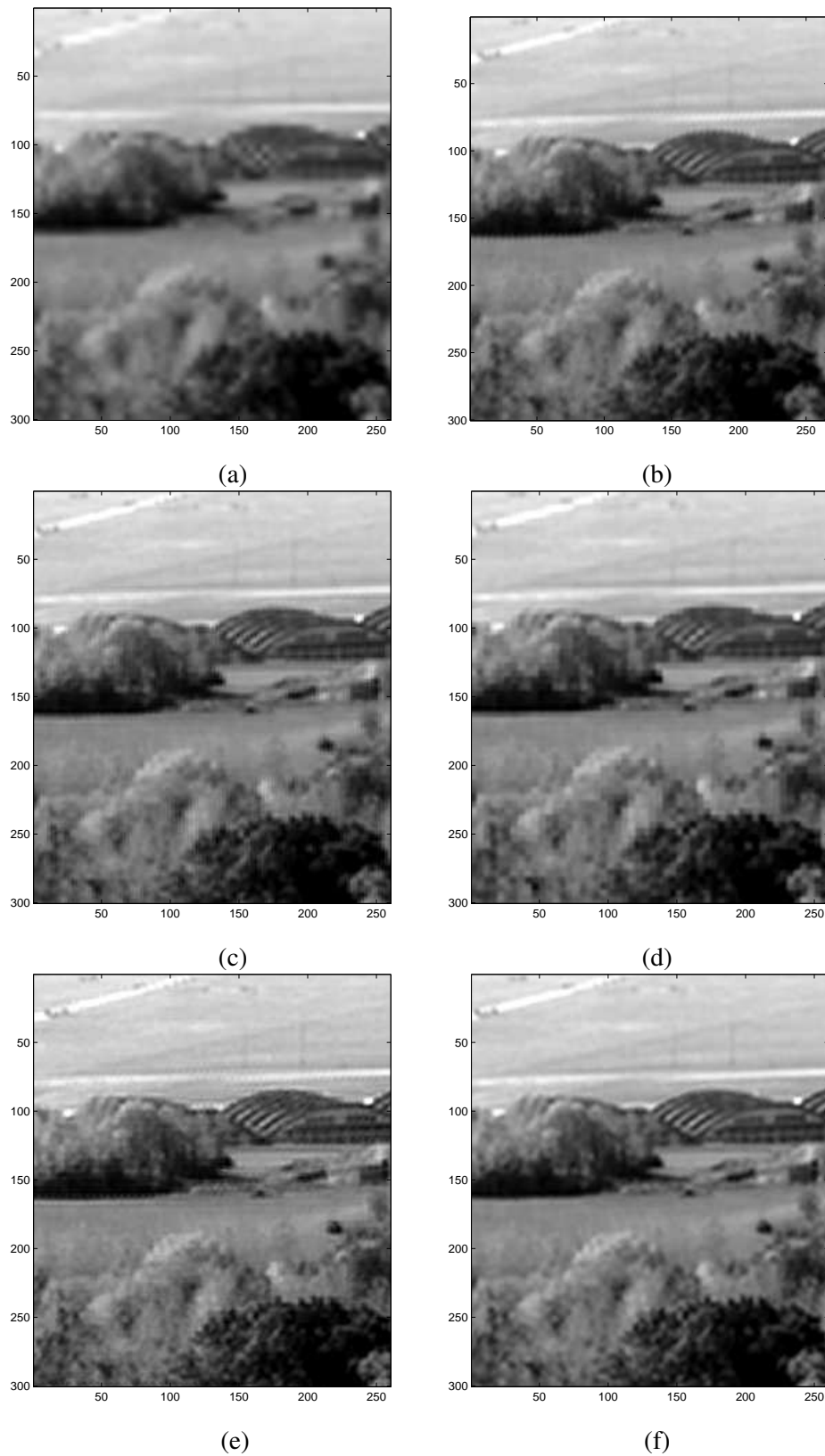


Fig. 16. Outputs for the FLIR image sequence using  $P = 16$  frames and  $L_x = L_y = 5$ . (a) Single frame bicubic interpolation, (b) SA+Wiener, (c) WNN with  $l = 4$ , (d) SKWF, (e) PWS with  $M = 10$ , and (f) AWF with  $Q = 20$ .

NSRs. While we have focused on translational motion, the framework proposed here can be applied to other motion, so long as the the PSF and motion models commute [13].

In the quantitative performance comparison conducted here with moderate levels of noise, the AWF using the spatially varying statistical model produces not only the lowest errors (MSE and MAE), but also the lowest error standard deviation over shift pattern realizations. This robustness to shift pattern makes the AWF attractive for video applications, where a variety of inter-frame motion patterns may be encountered over time. We believe that the images produced by the AWF also tend to have a subjectively pleasing appearance. In particular, the AWF with spatially varying statistics (i.e.,  $Q > 1$ ) will tend to do more smoothing in the low variance image regions that are dominated by noise, and more sharpening in the dynamic portion of the image. Note that the adaptive statistical model is most beneficial in cases where moderate to high noise levels exist. If no noise is present, the spatially varying model reduces to the global model, as can be seen in (19). One of the main advantages of the AWF SR algorithm over its predecessor, the PWS SR method, is that it does not quantize the motion parameters and place the LR pixels on a discrete HR grid. Also, the filter weights for the AWF SR method are model-based and do not require empirical training images. Finally, no vector quantization is needed. Instead, local NSRs are estimated for each observation window, allowing the AWF SR method to process the LR input frames in a spatially varying manner. This not only lowers the computational complexity for training and filtering, but also provides notably better performance in the experiments conducted here.

When considering translational motion, the computational complexity of the proposed method is quite low in comparison to iterative SR algorithms, such as the RLS [1]. In fact, for translational motion, the complexity is among the lowest of the published SR algorithms [4, 12, 16, 17] and can be readily adjusted based on several filter parameters. Like the methods in [16, 17], the AWF lends itself to parallel implementation as each observation window could potentially be processed independently in parallel. For non-translational motion the complexity goes up dramatically, but the option of a parallel implementation remains.

#### ACKNOWLEDGMENTS

The author would like to thank the Sensors Directorate at the Air Force Research Labs, Wright Patterson Air Force Base, for providing the FLIR video sequence used here. Helpful conversations with Dr. Raul Ordonez regarding statistical models are also greatly appreciated. Finally, the author would like to thank the anonymous reviewers for providing feedback to help strengthen this paper.

## REFERENCES

- [1] R. C. Hardie, K. J. Barnard, J. G. Bogner, E. E. Armstrong, and E. A. Watson, "High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system," *Opt. Eng.*, vol. 37, no. 1, pp. 247–260, Jan 1998.
- [2] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: A technical overview," *IEEE Signal Processing Mag.*, pp. 21–36, May 2003.
- [3] J. C. Gillette, T. M. Stadtmiller, and R. C. Hardie, "Reduction of aliasing in staring infrared imagers utilizing subpixel techniques," *Opt. Eng.*, vol. 34, no. 11, pp. 3130–3137, Nov 1995.
- [4] M. S. Alam, J. G. Bogner, R. C. Hardie, and B. J. Yasuda, "Infrared image registration using multiple translationally shifted aliased video frames," *IEEE Instrum. Meas. Mag.*, vol. 49, no. 5, Oct 2000.
- [5] S. Lertrattanapanich and N. K. Bose, "High resolution image formation from low resolution frames using delaunay triangulation," *IEEE Trans. Image Processing*, vol. 11, no. 12, pp. 1427–1441, Dec 2002.
- [6] K. Sauer and J. Allebach, "Iterative reconstruction of band-limited images from non-uniformly spaced samples," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 1497–1505, 1987.
- [7] K. Aizawa, T. Komatsu, and T. Saito, "Acquisition of very high resolution images using stereo camera," in *Proc. SPIE Visual Communication and Image Processing*, vol. 1, Boston, MA, Nov 1991, pp. 318–328.
- [8] A. Tekalp, M. Ozkan, and M. Sezan, "High resolution image reconstruction from lower-resolution image sequences and space-varying image restoration," in *Proc. ICASSP '92*, vol. 3, San Francisco, CA, Mar 1992, pp. 169–172.
- [9] A. Patti, M. Sezan, and A. Teklap, "Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time," *IEEE Trans. Image Processing*, vol. 6, no. 8, pp. 1064–1076, Aug 1997.
- [10] H. Shekarforoush and R. Chellappa, "Data-driven multi-channel super-resolution with application to video sequences," *J. Opt. Soc. Amer. A*, vol. 16, no. 3, pp. 481–492, Mar 1999.
- [11] N. Nguyen and P. Milanfar, "A wavelet-based interpolation restoration method for superresolution," *Circuits, Syst., Signal Process.*, vol. 19, no. 4, pp. 321–338, Aug 2000.
- [12] M. Elad and Y. Hel-Or, "A fast super-resolution reconstruction algorithm for pure translational motion and common space invariant blur," *IEEE Trans. Image Processing*, vol. 10, no. 8, pp. 1187–1193, Aug 2001.
- [13] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multi-frame super-resolution," *IEEE Trans. Image Processing*, vol. 13, no. 10, pp. 1327–1344, Oct 2004.
- [14] M. Shao, K. E. Barner, and R. C. Hardie, "Partition-based interpolation for image demosaicking and super-resolution reconstruction," *Opt. Eng.*, vol. 44, pp. 107003–1–107003–14, Oct 2005.
- [15] T. Q. Pham, L. J. van Vliet, and K. Schutte, "Robust fusion of irregularly sampled data using adaptive normalized convolution," *EURASIP Journal on Applied Signal Processing*, vol. 2006, no. Article ID 83268, pp. 1–12, 2006.
- [16] J. Shi, S. E. Reichenbach, and J. D. Howe, "Small-kernel superresolution methods for microscanning imaging systems," *Applied Optics*, vol. 45, no. 6, pp. 1203–1214, Feb 2006.
- [17] B. Narayanan, R. C. Hardie, K. E. Barner, and M. Shao, "A computationally efficient super-resolution algorithm for video processing using partition filters," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 5, pp. 621–634, May 2007.
- [18] K. E. Barner, A. M. Sarhan, and R. C. Hardie, "Partition-based weighted sum filters for image restoration," *IEEE Trans. Image Processing*, vol. 8, no. 5, pp. 740–745, May 1999.
- [19] M. Shao and K. E. Barner, "Optimization of partition-based weighted sum filters and their application to image denoising," *IEEE Trans. Image Processing*, Oct 2004, accepted for publication.

- [20] Y. Lin, R. C. Hardie, and K. E. Barner, "Subspace partition weighted sum filters for image restoration," *IEEE Signal Processing Letters*, vol. 12, no. 9, pp. 613–616, Sept 2005.
- [21] J.-S. Lee, "Digital image enhancement and noise filtering by use of local statistics," vol. 2, no. 3, Mar 1990.
- [22] J. S. Lim, *Two-Dimensional Signal and Image Processing*. Prentice Hall, 1990.
- [23] T. R. Tuinstra and R. C. Hardie, "High resolution image reconstruction from digital video by exploitation on non-global motion," *Opt. Eng.*, vol. 38, no. 5, May 1999.
- [24] J. Goodman, *Introduction to Fourier Optics*. McGraw-Hill, 1968.
- [25] M. Irani and S. Peleg, "Improving resolution by image registration," *CHIP: Graph. Models Image Process.*, vol. 53, no. 3, pp. 231–239, May 1991.
- [26] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, Vancouver, Aug 1981.
- [27] C. W. Therrian, *Discrete Random Signals and Statistical Signal Processing*. Prentice Hall, 1992.
- [28] A. K. Jain, *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.
- [29] S. Farsiu, D. Robinson, and P. Milanfar, *Resolution enhancement software*, [www.soe.ucsc.edu/~milanfar/SR-Software.htm](http://www.soe.ucsc.edu/~milanfar/SR-Software.htm), 2004.

## LIST OF FIGURES

1	Observation model relating a desired 2d continuous scene, $d(x, y)$ , with a set of corresponding LR frames. . . . .	4
2	Alternative observation model replacing the motion model and combining of LR frames by a single non-uniform sampling process. This is equivalent to the observation model in Fig. 1 for translational motion. It is also equivalent for rotational motion if the PSF is circularly symmetric. . . . .	6
3	Overview of the proposed SR algorithm. . . . .	7
4	High resolution grid showing the LR pixel locations for three frames (circles, triangles, and squares). The large box represents the span of an observation window (here shown to include $5 \times 5$ LR pixel spacings). The small box represents the estimation window containing a set of $6 \times 6$ HR pixels to estimate based on the LR pixels in the observation window. . . . .	8
5	Autocorrelation function model $r_{dd}(x, y)$ for $\sigma_d^2 = 1$ and $\rho = 0.75$ . . . . .	11
6	Scatter plot showing the empirical relationship between $\sigma_{f_i}$ and $\sigma_{d_i}$ based on the 8 bit aerial image in Fig. 9(b). For these data, $L_x = L_y = 4$ , $W_x = W_y = 12$ , and $h(x, y) = \frac{1}{16}\text{rect}(\frac{x}{4}, \frac{y}{4})$ , where $x$ and $y$ are measured in HR pixel spacings. . . . .	13
7	Weights for the case where $P = 3$ , $L_x = L_y = 3$ , $W_x = W_y = 9$ , and $h(x, y) = \frac{1}{9}\text{rect}(\frac{x}{3}, \frac{y}{3})$ , where $x$ and $y$ are measured in HR pixel spacings. (a) NSR = 1.0 (b) NSR = 0.1 . . . . .	14
8	Flop counts as a function of the HR image dimension $\sqrt{N}$ for both the proposed AWF SR and the WNN algorithm. For the AWF SR curves, $L_x = L_y = D_x = D_y = 4$ and $Q = 1$ . For the WNN, $l = 4$ . . . . .	16
9	Images used for error analysis with simulated video sequences. (a) Test image used to create an image sequence with translational motion, (b) training image used to estimate statistical parameters, (c) simulated LR frame with $L_x = L_y = 4$ and no noise, and (d) simulated LR frame with $L_x = L_y = 4$ and noise with variance $\sigma_n^2 = 100$ . . . . .	19
10	Translational shifts for 16 simulated LR frames. . . . .	20
11	Mean squared error as a function of the number of LR frames used for simulated image sequence with additive noise of variance $\sigma_n^2 = 100$ . . . . .	21
12	Mean absolute error as a function of the number of LR frames used for simulated image sequence with additive noise of variance $\sigma_n^2 = 100$ . . . . .	21



13	Output images for the simulated image sequence with $P = 16$ . (a) Single frame bicubic interpolation, (b) SA + Wiener, (c) WNN with $l = 4$ , (d) SKWF, (e) PWS with $M = 10$ , and (f) AWF with $Q = 20$ . . . . .	22
14	Single $64 \times 64$ frame in the FLIR image sequence. . . . .	25
15	Estimated translational shifts (in HR pixel spacings) for the FLIR image sequence. . . . .	25
16	Outputs for the FLIR image sequence using $P = 16$ frames and $L_x = L_y = 5$ . (a) Single frame bicubic interpolation, (b) SA+Wiener, (c) WNN with $l = 4$ , (d) SKWF, (e) PWS with $M = 10$ , and (d) AWF with $Q = 20$ . . . . .	26

## LIST OF TABLES

I	MSE and MAE statistics for the various SR algorithms over 100 shift pattern realizations. . .	23
---	---	----