IRWIN AND JOAN JACOBS
CENTER FOR COMMUNICATION AND INFORMATION TECHNOLOGIES

A Hybrid Vector Wiener Filter Approach to Translational Super-Resolution

Tomer Michaeli and Yonina C. Eldar

CCIT Report #756
February 2010

DEPARTMENT OF ELECTRICAL ENGINEERING
TECHNION - ISRAEL INSTITUTE OF TECHNOLOGY, HAIFA 32000, ISRAEL

CCIT REPORT #756 February 2010

A Hybrid Vector Wiener Filter Approach to Translational Super-Resolution

Tomer Michaeli, *Student Member, IEEE* and Yonina C. Eldar, *Senior Member, IEEE*

***Abstract*—We address the problem of purely-translational super-resolution (SR) for signals in arbitrary dimensions. We show that discretization, a key step in many SR algorithms, inevitably leads to inaccurate modeling. Instead, we treat the problem entirely in the continuous domain by modeling the signal as a continuous-space random process and deriving its linear minimum mean-squared error (LMMSE) estimate given the low-resolution discrete-space observations. We derive a closed form expression for the resulting mean-squared error and use it to analyze the emergence of periodic artifacts in the super-resolved signal. We also provide three efficient implementation schemes of the LMMSE estimate, one of which specialized for 1D applications. These methods constitute a natural generalization of several well known single-image recovery algorithms, such as spline interpolation, to the multichannel SR setting. Experiments on real-world images demonstrate the advantage of our approach with respect to several prominent SR techniques that rely on discretization.**

***Index Terms*—Super-resolution, nonuniform interpolation, hybrid Wiener filter.**

I. Introduction

SUPER-RESOLUTION refers to the process of combining several low-resolution descriptions of a signal to form one higher resolution version of it. In the field of image processing, there are 1D, 2D and 3D variants of this task. Perhaps the most commonly treated scenario is that of purely-translational spatial (2D) super-resolution (SR) [1], [2]. Here, several low-resolution noisy images of a scene are captured by a camera, each with a different translation, and the goal is to produce one high-resolution image of the same scene. Three-dimensional scenarios arise in space-time SR applications [3]. There, several video sequences of the same scene are fused into one higher-resolution video stream at a higher frame-rate. If the video cameras are spatially co-calibrated, then processing can be carried out only along the time dimension (temporal SR), rendering the problem one-dimensional. In this paper, we collectively refer to all problems of this type as SR, and develop a theory for arbitrary d-dimensional signals.

The physical model underlying purely-translational SR scenarios can be described mathematically as [4]

$$c_k[\boldsymbol{n}] = [(s * x)(\boldsymbol{t})]_{\boldsymbol{t}=\boldsymbol{n}-\boldsymbol{t}_k} + u_k[\boldsymbol{n}], \qquad k = 1, \ldots, K, \quad (1)$$

where $c_k[\boldsymbol{n}]$, $\boldsymbol{n} \in \mathbb{Z}^d$, is the kth discrete-space observation of the continuous-space signal $x(\boldsymbol{t})$, $\boldsymbol{t} \in \mathbb{R}^d$, acquired with translation $\boldsymbol{t}_k$ and additive noise $u_k[\boldsymbol{n}]$. The filter $s(\boldsymbol{t})$ is associated with the imaging device. It corresponds to the point-spread-function (PSF) of the lens in 2D SR, to the temporal integration profile of the sensor in 1D SR, and to a combination of both in 3D space-time SR. The goal of an SR algorithm is to produce a discrete-space high-resolution signal $\boldsymbol{x}_{\text{HR}}[\boldsymbol{n}]$ which corresponds to the samples of $x(\boldsymbol{t})$ on a dense grid.

In much of the recent literature, SR is modeled via the discrete-space relations [1], [2]:

$$\boldsymbol{c}_k = \boldsymbol{S}_k \boldsymbol{x}_{\text{HR}} + \boldsymbol{u}_k, \qquad k = 1, \ldots, K. \quad (2)$$

Here $\boldsymbol{c}_k$, $\boldsymbol{u}_k$, and $\boldsymbol{x}_{\text{HR}}$ are column vectors comprising the elements of $c_k[\boldsymbol{n}]$, $u_k[\boldsymbol{n}]$, and $\boldsymbol{x}_{\text{HR}}[\boldsymbol{n}]$ respectively, and the matrix $\boldsymbol{S}_k$ accounts for the filtering by $s(\boldsymbol{t})$ and the sampling. This discrete formulation has several advantages over the continuous model (1). Most noteworthy, it allows the construction of a finite-dimensional optimization problem incorporating complicated realistic prior-knowledge assumptions on the unknown $\boldsymbol{x}_{\text{HR}}$ [2]. Such optimization problems may be solved using standard optimization methods. Nevertheless, as we show in this paper, there are many situations in which the continuous-space equations (1) cannot be represented in the discrete form (2). In other words, we show that in certain settings SR must be regarded as a continuous-space interpolation problem. Algorithms which do not treat the problem from this viewpoint, inevitably suffer from a model miss-match error.

Prior SR work falling into the interpolation category (also termed frequency-domain methods [4]) include [5], [6], [7], [8], [9], [10], [11], [12]. In these algorithms, the continuous-space scene $x(\boldsymbol{t})$ is assumed to be bandlimited. In the single-channel interpolation literature, recent work has shown that this type of prior knowledge is often not in agreement with the typical behavior of natural images [13], [14]. Furthermore, it has been experimentally demonstrated that interpolation algorithms relying on non-bandlimited models, such as the Matèrn prior, can lead to improved reconstruction results [15], [16], [17]. In SR applications, the bandlimited assumption translates to a stringent limitation on the required number of measurements. Specifically, the number of observations K required for increasing the resolution by a factor of Δ must be at least Δ^d in the methods mentioned above. Thus, to increase resolution by a factor of 4 in every dimension, for example, at least 16 still images are required in a spatial SR task ($d = 2$), and at least 64 video sequences are needed in a space-time SR scenario ($d = 3$).

This work was supported in part by the Israel Science Foundation under Grant no. 1081/07 and by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMmunications NEWCOM++ (contract no. 216715).

The authors are with the department of electrical engineering, Technion–Israel Institute of Technology, Haifa, Israel (phone: +972-4-8294682, fax: +972-4-8295757, e-mail: tomermic@tx.technion.ac.il, yonina@ee.technion.ac.il).

An additional disadvantage of existing continuous-domain SR methods is that they do not explicitly incorporate statistical assumptions on $x(\boldsymbol{t})$. Such assumptions have been shown to greatly improve SR performance when using the discrete model (2) [2].

In [18], the authors proposed an SR approach, which can (in theory) be adapted to work in the continuous-domain, and which does not rely on the bandlimited assumption. There, prior knowledge on $x(\boldsymbol{t})$ is implicitly defined by a user-chosen "back-projection kernel". However, the effect of this kernel on the reconstruction is not clear. Moreover, by adapting this algorithm to the continuous-domain, we end up with the need to iteratively update the continuous-space recovery $\hat{x}(\boldsymbol{t})$. This, of course, cannot be done in a precise manner. In practice, only the densely sampled version $\hat{\boldsymbol{x}}_{\text{HR}}[\boldsymbol{n}]$ is maintained throughout the iterations. Therefore, practically, this method belongs to the category of algorithms relying on the incorrect discrete formulation (2).

In this paper we treat the SR interpolation problem without resorting to the commonly used bandlimited assumption. Motivated by the good single-channel image interpolation results reported in [13], [14], [15], [16], [17], here we model $x(\boldsymbol{t})$ as a wide-sense-stationary (WSS) continuous-space random signal with known power-spectral-density (PSD). We then derive the linear minimum mean-squared error (MSE) estimator of $x(\boldsymbol{t})$ given the discrete-space observations $c_1[\boldsymbol{n}], \ldots, c_K[\boldsymbol{n}]$. We term this estimator the *vector hybrid Wiener filter* as it operates on a discrete-space vector process $\boldsymbol{c}[\boldsymbol{n}] = (c_1[\boldsymbol{n}], \ldots, c_K[\boldsymbol{n}])^T$, and outputs a continuous-space recovery $\hat{x}(\boldsymbol{t})$.

We show that in situations where both the PSF and the signal's autocorrelation function are compactly supported, the resulting estimator can be implemented efficiently. Specifically, for 1D SR, we provide an implementation which consists of digital filtering of the discrete-space measurements $c_k[\boldsymbol{n}]$, followed by a simple interpolation stage. For SR in higher dimensions (*e.g.,* spatial or space-time SR), we propose two algorithms for efficient implementation of the resulting estimator. Both approaches include an iterative stage which operates on the measurements $c_1[\boldsymbol{n}], \ldots, c_K[\boldsymbol{n}]$, followed by an interpolation step. An important feature of our iterative approaches, is that iterations are performed in the digital domain so that the implementation stays loyal to the continuous formulation (1). In other words, our approach does not rely on approximations of the type required to adapt [18] to work in the continuous-domain.

When the autocorrelation function of the scene is chosen to be a B-spline, our approach constitutes a natural generalization of the widely used (single-channel) spline interpolation method to the multichannel SR setting. We term the resulting algorithm *spline super resolution*. As we demonstrate through experiments on real-world images, our multichannel spline interpolation method often yields recoveries, which are better than those produced by current state-of-the-art algorithms.

The paper is organized as follows. In Section II, we show mathematically why discretization is generally not loyal to the true physical setting. In Section III, we present our assumed continuous-space statistical model and derive the corresponding LMMSE estimator. Three efficient implementation schemes for the LMMSE estimator are provided in Sections IV and V. In Section VI, we analyze the reconstruction error patterns and discuss the inherent tradeoff between a sharp recovery and periodic artifacts. To obtain a practical SR algorithm, we specialize our approach in Section VII to spline autocorrelation models, which leads to the spline super resolution algorithm. Finally, in Section VIII we present several experiments on real-world images, demonstrating the advantage of our approach over SR algorithms which rely on discretization.

II. The Need for Continuous-Space Treatment

In this section we analyze the situations in which purely-translational SR cannot be accurately modeled via the discrete form (2).

A. Motivation

To develop intuition, we begin by examining a simple case of an imaging device whose lens' PSF is negligible with respect to pixel size. In this situation, the pixels in the image correspond to integrals of the continuous-space scene $x(\boldsymbol{t})$ over non-overlapping rectangular environments. Therefore, the filter $s(\boldsymbol{t})$ of (1) in this case is a rectangular kernel:

$$s(\boldsymbol{t}) = \begin{cases} 1 & \boldsymbol{t} \in [0,1]^d \\ 0 & \text{else.} \end{cases} \tag{3}$$

Now, assume we wish to construct a representation $x_{\text{HR}}[\boldsymbol{n}]$ whose resolution is Δ times higher in each dimension. In other words, we want the pixels in $x_{\text{HR}}[\boldsymbol{n}]$ to correspond to integrations of $x(\boldsymbol{t})$ over contiguous non-overlapping environments of size $1/\Delta$. This setting is demonstrated in Fig. 1 for a uni-dimensional signal ($d = 1$), with $t_1 = 0$, $t_2 = -1/2$ and $\Delta = 2$. As can be seen, in this case each of the elements in the low-resolution sequences $c_k[n]$ can be expressed as a linear combination of the high-resolution pixels $x_{\text{HR}}[n]$. For example, $c_1[1] = x_{\text{HR}}[1] + x_{\text{HR}}[2]$, and similarly $c_2[3] = x_{\text{HR}}[4] + x_{\text{HR}}[5]$.

The above example demonstrates that in some situations, the discrete formulation (2) is loyal to the physical settings. However, what made discretization possible in this scenario was the fact that the magnification Δ was an integer (namely $\Delta = 2$), and that the translations t_k were integer multiples of $1/\Delta$ (namely 0 and $-1/2$). It is easily verified that unless these two conditions are met, discretization is impossible in this setting.

Typically, the magnification factor Δ is indeed chosen to be an integer. Furthermore, one may argue that when Δ is sufficiently large, the translation $\boldsymbol{t}_k$ can be quite accurately approximated by $\boldsymbol{\ell}_k/\Delta$ for some $\boldsymbol{\ell}_k \in \mathbb{Z}^d$, so that use of the discrete model (2) is justified. However, as we show next, even under these two assumptions, discretization is generally *impossible* for non-rectangular PSF functions.

B. Discretization with General PSFs

To study discretization with arbitrary PSF functions, we need to specify explicitly the relation between the discrete-space high resolution signal $x_{\text{HR}}[\boldsymbol{n}]$ and the continuous-space

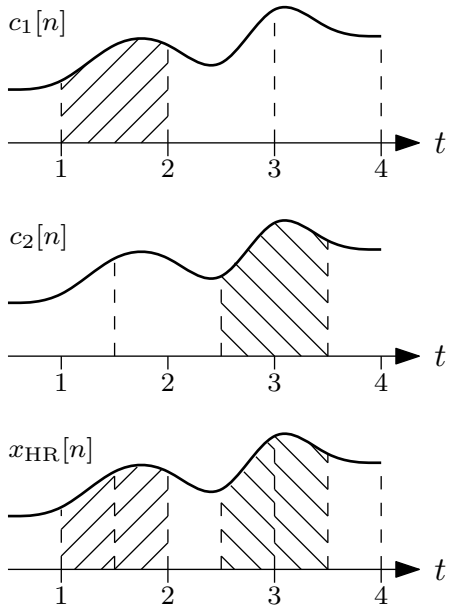


Fig. 1: Increasing resolution by a factor of 2 with a rectangular PSF. The low resolution sequences $c_1[n]$ and $c_2[n]$ were acquired with translations $t_1 = 0$ and $t_2 = -1/2$ respectively, and correspond to linear combinations of elements from the high resolution sequence $x_{\mathrm{HR}}[n]$.

signal $x(\boldsymbol{t})$. Here we assume that $x_{\mathrm{HR}}[\boldsymbol{n}]$ corresponds to the samples of $x(\boldsymbol{t})$ on a grid of points $\{\boldsymbol{n}/\Delta : \boldsymbol{n} \in \mathbb{Z}^d\}$ for some magnification $\Delta > 1$, after having been convolved with a PSF $w(\boldsymbol{t})$:

$$x_{\mathrm{HR}}[\boldsymbol{n}] = [(w * x)(\boldsymbol{t})]_{\boldsymbol{t}=\boldsymbol{n}/\Delta}\,. \tag{4}$$

This formulation is general enough to capture a wide range of interesting scenarios. For example, by choosing the target PSF $w(\boldsymbol{t}) = \delta(\boldsymbol{t})$, we get that $x_{\mathrm{HR}}[\boldsymbol{n}]$ corresponds to the pointwise samples $x(\boldsymbol{n}/\Delta)$. The PSF $w(\boldsymbol{t})$ can also be chosen to provide $\boldsymbol{x}_{\mathrm{HR}}[\boldsymbol{n}]$ with the realistic impression that it was obtained by moving the imaging device towards the scene. This is done by setting $w(\boldsymbol{t}) = s(\Delta \boldsymbol{t})$, as in the example of Fig. 1. Finally, if the the profile of $s(\boldsymbol{t})$ is too wide with respect to the low-resolution grid, then $w(\boldsymbol{t})$ can be set to be narrower than $s(\Delta \boldsymbol{t})$, in order to obtain a deblurring effect in the high-resolution domain.

Using (4), $x_{\mathrm{HR}}[\boldsymbol{n}]$ can be written as

$$x_{\mathrm{HR}}[\boldsymbol{n}] = \int_{\mathbb{R}^d} x(\boldsymbol{t})\, w\Big(\frac{\boldsymbol{n}}{\Delta} - \boldsymbol{t}\Big)\, d\boldsymbol{t}. \tag{5}$$

Similarly, (1) implies that the low-resolution samples $c_k[\boldsymbol{n}]$ are given by

$$c_k[\boldsymbol{n}] = \int_{\mathbb{R}^d} x(\boldsymbol{t}) s(\boldsymbol{n} - \boldsymbol{t} - \boldsymbol{t}_k) d\boldsymbol{t}. \tag{6}$$

The conjecture that $c_k[\boldsymbol{n}]$ is a linear function of $x_{\mathrm{HR}}[\boldsymbol{n}]$ corresponds to the existence of a set of coefficients $\boldsymbol{S}_k[\boldsymbol{m}, \boldsymbol{n}]$, $\boldsymbol{m}, \boldsymbol{n} \in \mathbb{Z}^d$, such that

$$\begin{aligned} c_k[\boldsymbol{n}] &= \sum_{\boldsymbol{m}\in\mathbb{Z}^d} \boldsymbol{S}_k[\boldsymbol{n}, \boldsymbol{m}] x_{\mathrm{HR}}[\boldsymbol{m}] \\ &= \sum_{\boldsymbol{m}\in\mathbb{Z}^d} \boldsymbol{S}_k[\boldsymbol{n}, \boldsymbol{m}] \int_{\boldsymbol{t}\in\mathbb{R}^d} x(\boldsymbol{t})\, w\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{t}\Big)\, d\boldsymbol{t} \\ &= \int_{\boldsymbol{t}\in\mathbb{R}^d} x(\boldsymbol{t}) \left(\sum_{\boldsymbol{m}\in\mathbb{Z}^d} \boldsymbol{S}_k[\boldsymbol{n}, \boldsymbol{m}]\, w\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{t}\Big) \right) d\boldsymbol{t}, \end{aligned} \tag{7}$$

where we used (5). Discretization is possible if and only if (7) equals (6) for every signal $x(\boldsymbol{t})$ in $L_2(\mathbb{R}^d)$. This happens only if

$$s(\boldsymbol{n} + \boldsymbol{t} - \boldsymbol{t}_k) = \sum_{\boldsymbol{m}\in\mathbb{Z}^d} \boldsymbol{S}_k[\boldsymbol{n}, \boldsymbol{m}]\, w\Big(\frac{\boldsymbol{m}}{\Delta} + \boldsymbol{t}\Big) \tag{8}$$

for every $\boldsymbol{n} \in \mathbb{Z}^d$ and for almost every $\boldsymbol{t} \in \mathbb{R}^d$.

Taking the d-dimensional continuous-space Fourier transform (CSTF) of both sides of (8) with respect to $\boldsymbol{t}$, yields the condition

$$S(\boldsymbol{\omega}) \mathrm{e}^{\mathrm{i}\boldsymbol{\omega}^T(\boldsymbol{n}-\boldsymbol{t}_k)} = W(\boldsymbol{\omega}) A_{k,n}(\mathrm{e}^{-\mathrm{i}\boldsymbol{\omega}/\Delta}). \tag{9}$$

Here $\boldsymbol{\omega}$ denotes the d-dimensional frequency vector, $S(\boldsymbol{\omega})$ and $W(\boldsymbol{\omega})$ are the CSTFs of $s(\boldsymbol{t})$ and $w(\boldsymbol{t})$ respectively, and $A_{k,n}(\mathrm{e}^{\mathrm{i}\boldsymbol{\omega}})$ is the discrete-space Fourier transform (DSFT) of $\boldsymbol{S}_k[\boldsymbol{n}, \boldsymbol{m}]$ with respect to the index $\boldsymbol{m}$. As motivated in Section II-A, we now make the simplifying assumption that the magnification factor Δ is an integer, and that the shift vector $\boldsymbol{t}_k$ comprises only integer multiples of $1/\Delta$. We therefore substitute $\boldsymbol{t}_k = \boldsymbol{\ell}_k/\Delta$, where $\boldsymbol{\ell}_k$ is some vector in $\mathbb{Z}^d$, and get that $S(\boldsymbol{\omega})$ and $W(\boldsymbol{\omega})$ must satisfy

$$S(\boldsymbol{\omega}) = W(\boldsymbol{\omega}) A_{k,n}(\mathrm{e}^{-\mathrm{i}\boldsymbol{\omega}/\Delta}) \mathrm{e}^{\mathrm{i}\boldsymbol{\omega}^T(\boldsymbol{\ell}_k - \Delta\boldsymbol{n})/\Delta}. \tag{10}$$

Finally, noting that $A_{k,n}(\mathrm{e}^{-\mathrm{i}\boldsymbol{\omega}/\Delta}) \mathrm{e}^{\mathrm{i}\boldsymbol{\omega}^T(\boldsymbol{\ell}_k - \Delta\boldsymbol{n})/\Delta}$ is a function which is $2\pi\Delta$-periodic in each of the elements of the frequency vector $\boldsymbol{\omega}$, we arrive at the following conclusion.

Proposition 1: *Let $S(\boldsymbol{\omega})$ and $W(\boldsymbol{\omega})$ denote the CSFTs of the two PSFs $s(\boldsymbol{t})$ and $w(\boldsymbol{t})$ respectively. Furthermore, let $\Delta \geq 1$ be an integer magnification factor and $\boldsymbol{t}_k$ be a shift vector whose elements are integer multiples of $1/\Delta$. Assume that a continuous-space scene $x(\boldsymbol{t})$ is sampled on the grid $\{\boldsymbol{t} = \boldsymbol{n} - \boldsymbol{t}_k : \boldsymbol{n} \in \mathbb{Z}^d\}$, after having been convolved with $s(\boldsymbol{t})$, to yield the discrete-space signal $c_k[\boldsymbol{n}]$. Similarly, assume that $x(\boldsymbol{t})$ is sampled on the grid $\{\boldsymbol{t} = \boldsymbol{n}/\Delta : \boldsymbol{n} \in \mathbb{Z}^d\}$ after having been convolved with $w(\boldsymbol{t})$, to yield the discrete-space signal $x_{HR}[\boldsymbol{n}]$. Then $c_k[\boldsymbol{n}]$ can be written as a linear function of $x_{HR}[\boldsymbol{n}]$* for every scene $x(\boldsymbol{t})$ *if and only if there exists a $2\pi\Delta$-periodic function $B_k(\mathrm{e}^{\mathrm{i}\boldsymbol{\omega}/\Delta})$ such that*

$$S(\boldsymbol{\omega}) = W(\boldsymbol{\omega}) B_k(\mathrm{e}^{\mathrm{i}\boldsymbol{\omega}/\Delta}) \tag{11}$$

for almost every $\boldsymbol{\omega} \in [-\pi, \pi]^d$.

We see that even in the simplistic case of integer magnification and quantized shifts, the target PSF $w(\boldsymbol{t})$ cannot be arbitrary. To demonstrate the implications of Proposition 1,

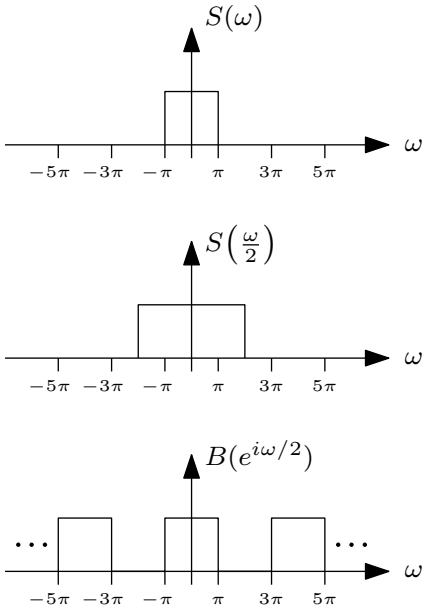


Fig. 2: Increasing resolution by a factor of 2 with the PSF $s(t) = \mathrm{sinc}(t)$. Here $S(\omega) = \mathrm{rect}(\omega/(2\pi))$ (top) and consequently $S(\omega/2)$ (middle) can be expressed as the product of $S(\omega)$ and a rectangular wave of period 4π (bottom).

consider the case in which $w(\boldsymbol{t}) = s(\Delta \boldsymbol{t})$. In this situation, condition (12) becomes

$$S(\boldsymbol{\omega}) = S\Big(\frac{\boldsymbol{\omega}}{\Delta}\Big)\, B_k(\mathrm{e}^{\mathrm{i}\boldsymbol{\omega}/\Delta}). \tag{12}$$

As we have seen in Section II-A, this condition holds for a rectangular PSF. As another example, it is easy to verify that (12) is satisfied when the PSF $s(\boldsymbol{t})$ is an ideal low-pass filter $\mathrm{sinc}(\boldsymbol{t})$, as demonstrated in Fig. 2. However, in this scenario, the entire frequency content above $\omega = \pi$ is zeroed out in each dimension, rendering super-resolution ill-posed. A more realistic situation corresponds to the Gaussian PSF model, which is widely used in the SR literature. In this case,

$$S(\boldsymbol{\omega}) = \exp\left\{\frac{-\|\boldsymbol{\omega}\|^2}{2\sigma_s^2}\right\}. \tag{13}$$

for some $\sigma_s > 0$. Direct computation shows that

$$\frac{S(\boldsymbol{\omega})}{S(\frac{\boldsymbol{\omega}}{\Delta})} = \exp\left\{\frac{-\|\boldsymbol{\omega}\|^2}{2\sigma_s^2}\left(1 - \frac{1}{\Delta^2}\right)\right\}, \tag{14}$$

which is not a $2\pi\Delta$-periodic function. Therefore this kind of PSF does not conform with (12), and cannot be treated via the discrete model (2). We conclude that to be loyal to the physical setting when the conditions of Proposition 1 are not met, SR must be addressed in the continuous domain.

III. The Vector Hybrid Wiener Filter

The discussion above revealed that the discrete model (2) is generally invalid. Therefore, we now turn to the problem of recovering the high resolution description $x_{\mathrm{HR}}[\boldsymbol{n}]$ from the low resolution sequences $c_k[\boldsymbol{n}]$, $k = 1, \ldots, K$, based on the continuous-space model (1).

A. Statistical Model

Ultimately, our goal is to devise an algorithm which produces an estimate $\hat{x}_{\mathrm{HR}}[\boldsymbol{n}]$ minimizing the MSE

$$\epsilon_x^2[\boldsymbol{n}] = E\Big[\,(\hat{x}_{\mathrm{HR}}[\boldsymbol{n}] - x_{\mathrm{HR}}[\boldsymbol{n}])^2 \Big|\, x(\boldsymbol{t})\Big]\,, \tag{15}$$

for every pixel $\boldsymbol{n} \in \mathbb{Z}^d$, where the expectation is over realizations of the noise sequences $u_k[\boldsymbol{n}]$, $k = 1 \ldots, K$ in (1). Unfortunately, the MSE depends on the underlying scene because $x_{\mathrm{HR}}[\boldsymbol{n}]$ is a function of $x(\boldsymbol{t})$, as implied by (5). Minimizing $\epsilon_x^2[\boldsymbol{n}]$ uniformly over all signals $x(\boldsymbol{t})$ is impossible. Consequently, comparison between different SR approaches is not a well defined problem since one method may be better than another for some signals $x(\boldsymbol{t})$ and worse for others. To overcome this obstacle, we propose modeling the signal $x(\boldsymbol{t})$ as the realization of a random process with known statistics. This allows the replacement of the signal-dependent MSE $\epsilon_x^2[\boldsymbol{n}]$ by its expectation $\epsilon^2[\boldsymbol{n}] = E[\epsilon_x^2[\boldsymbol{n}]]$ over all possible signal realizations, resulting in the criterion

$$\epsilon^2[\boldsymbol{n}] = E\Big[(\hat{x}_{\mathrm{HR}}[\boldsymbol{n}] - x_{\mathrm{HR}}[\boldsymbol{n}])^2\Big]\,. \tag{16}$$

In the SR literature, various statistical priors have been proposed for describing the typical behavior of natural images. These include, for example, Gaussian and Huber random Markov fields [19], [20], [21]. However, these studies modeled the statistics of the desired discrete-space signal $x_{\mathrm{HR}}[\boldsymbol{n}]$ rather than that of the continuous-space signal $x(\boldsymbol{t})$. In the single image interpolation literature, several recent works have used the assumption that continuous-space natural scenes are realizations of some stationary process [13], [14], [15], [16], [17]. It has been demonstrated that when the power-spectral density (PSD) of $x(\boldsymbol{t})$ is chosen to have a polynomial decay in frequency, the resulting recovery is superior to the Shannon interpolation, which relies on the traditional bandlimited model. Based on these findings, in this paper we adopt the stationarity assumption and extend the Bayesian single image recovery techniques to the multi-frame SR scenario.

B. Optimal Linear Recovery

Our goal is to *linearly* estimate $x_{\mathrm{HR}}[\boldsymbol{n}]$ given the measurements $c_1[\boldsymbol{n}], \ldots, c_K[\boldsymbol{n}]$. Due to the linearity of the convolution operation in (4), the linear minimum MSE (LMMSE) estimate of $x_{\mathrm{HR}}[\boldsymbol{n}]$ is given by

$$\hat{x}_{\mathrm{HR}}[\boldsymbol{n}] = [(w * \hat{x})(\boldsymbol{t})]_{\boldsymbol{t}=\boldsymbol{n}/\Delta}\,, \tag{17}$$

where $\hat{x}(\boldsymbol{t})$ is the LMMSE estimate of $x(\boldsymbol{t})$ given the measurements. Therefore, to obtain a closed form expression for $\hat{x}_{\mathrm{HR}}[\boldsymbol{n}]$, we will first compute the continuous-space LMMSE recovery $\hat{x}(\boldsymbol{t})$ and then sample it on the high-resolution grid using (17).

Before analyzing our estimation problem in detail, we would like to note that the purely translational SR setting treated in this paper can be viewed as a special case of multichannel sampling, as schematically shown in Fig. 3. In this setting, a signal $x(\boldsymbol{t})$ passes through K filters, which in our case correspond to $s_k(\boldsymbol{t}) = s(\boldsymbol{t} - \boldsymbol{t}_k)$, $k = 1, \ldots, K$, contaminated

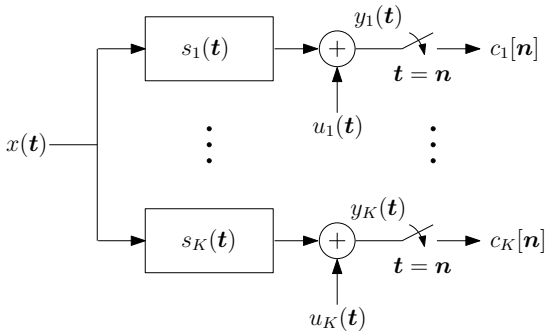


Fig. 3: Multichannel sampling scheme.

by continuous-space noise processes[1] $u_1(\boldsymbol{t}),\ldots,u_K(\boldsymbol{t})$, and then sampled on the grid $\{\boldsymbol{t}=\boldsymbol{n}:\boldsymbol{n}\in\mathbb{Z}^d\}$ to yield the observed measurements $c_1[\boldsymbol{n}],\ldots,c_K[\boldsymbol{n}]$. Thus, denoting

$$y_k(\boldsymbol{t})=(x*s_k)(\boldsymbol{t})+u_k(\boldsymbol{t}),\quad k=1,\ldots,K, \tag{18}$$

our goal can be more generally described as estimating a continuous-space signal $x(\boldsymbol{t})$ based on equidistant samples of a set of continuous-space processes $y_1(\boldsymbol{t}),\ldots,y_K(\boldsymbol{t})$, which are statistically related to $x(\boldsymbol{t})$, such that the MSE $E[(x(\boldsymbol{t})-\hat{x}(\boldsymbol{t})^2]$ is minimized for every $\boldsymbol{t}\in\mathbb{R}^d$.

The case in which only one measurement channel is available (*i.e.,* $K=1$) was treated in [22]. We refer to the resulting reconstruction formula as the scalar hybrid Wiener filter since its input is the (scalar) discrete-space signal $y_1(\boldsymbol{n})$, $\boldsymbol{n}\in\mathbb{Z}^d$, whereas its output is a continuous-space signal $\hat{x}(\boldsymbol{t})$, $\boldsymbol{t}\in\mathbb{R}^d$. Consequently, we refer to our multichannel setup (for $K>1$) as the *vector hybrid Wiener filter*.

To make the derivation general, all we assume in this section is that $x(\boldsymbol{t})$ and $y_1(\boldsymbol{t}),\ldots,y_K(\boldsymbol{t})$ are jointly WSS. We denote their cross-correlation functions by

$$R^k_{xy}(\boldsymbol{\tau})=E[x(\boldsymbol{t})y_k(\boldsymbol{t}-\boldsymbol{\tau})],\quad k=1,\ldots,K, \tag{19}$$
$$R^{k,\ell}_{yy}(\boldsymbol{\tau})=E[y_\ell(\boldsymbol{t})y_k(\boldsymbol{t}-\boldsymbol{\tau})],\quad k,\ell=1,\ldots,K. \tag{20}$$

The cross-spectra are given by the CSFTs

$$\Gamma^k_{xy}(\boldsymbol{\omega})=\mathcal{F}\{R^k_{xy}\}(\boldsymbol{\omega}),\quad k=1,\ldots,K, \tag{21}$$
$$\Gamma^{k,\ell}_{yy}(\boldsymbol{\omega})=\mathcal{F}\{R^{k,\ell}_{yy}\}(\boldsymbol{\omega}),\quad k,\ell=1,\ldots,K. \tag{22}$$

At any given location $\boldsymbol{t}\in\mathbb{R}^d$, the estimate $\hat{x}(\boldsymbol{t})$ corresponds to a linear combination of the measurements $y_1(\boldsymbol{n}),\ldots,y_K(\boldsymbol{n})$, $\boldsymbol{n}\in\mathbb{Z}^d$. Therefore, $\hat{x}(\boldsymbol{t})$ can be expressed as

$$\hat{x}(\boldsymbol{t})=\sum_{k=1}^{K}\sum_{\boldsymbol{n}\in\mathbb{Z}^d}y_k(\boldsymbol{n})\tilde{v}_k(\boldsymbol{t}-\boldsymbol{n}), \tag{23}$$

for some set of functions $\tilde{v}_1(\boldsymbol{t}),\ldots,\tilde{v}_k(\boldsymbol{t})$, which we call *interpolation kernels*. The following theorem provides a closed form for the set of interpolation kernels minimizing the MSE.

Theorem 2 (Vector hybrid Wiener filter): *The interpolation kernels $\tilde{v}_1(\boldsymbol{t}),\ldots,\tilde{v}_K(\boldsymbol{t})$ in (23) that minimize the MSE $E[(x(\boldsymbol{t})-\hat{x}(\boldsymbol{t}))^2]$ for every $\boldsymbol{t}\in\mathbb{R}^d$, are given in the frequency domain by*

$$\tilde{\boldsymbol{V}}(\boldsymbol{\omega})=\left(\sum_{\boldsymbol{n}\in\mathbb{Z}^d}\boldsymbol{\Gamma}_{yy}(\boldsymbol{\omega}-2\pi\boldsymbol{n})\right)^{-1}\boldsymbol{\Gamma}_{xy}(\boldsymbol{\omega}), \tag{24}$$

where $\boldsymbol{\Gamma}_{xy}(\boldsymbol{\omega})=(\Gamma^1_{xy}(\boldsymbol{\omega}),\ldots,\Gamma^K_{xy}(\boldsymbol{\omega}))^T$, $\tilde{\boldsymbol{V}}(\boldsymbol{\omega})=(\tilde{V}_1(\boldsymbol{\omega}),\ldots,\tilde{V}_K(\boldsymbol{\omega}))^T$ is a $K\times1$ vector comprising the CSFTs of the interpolation kernels, and $\boldsymbol{\Gamma}_{yy}(\boldsymbol{\omega})$ is a $K\times K$ matrix whose (k,ℓ)th entry is $\Gamma^{k,\ell}_{yy}(\boldsymbol{\omega})$.

Proof: See Appendix A. ∎

In the case $K=1$, (24) reduces to the scalar hybrid Wiener filter [22], which was used for the task of single-image zooming in [15], [16], [17].

In the multichannel setting $K>1$, the interpolation formula (23) can be broken up into K single-channel interpolation procedures. Specifically, (23) implies that each of the K low-resolution signals has to be interpolated using an appropriate reconstruction kernel and then the resulting K high-resolution estimates should be summed up to produce $\hat{x}(\boldsymbol{t})$. However, this naive implementation of (23) is typically not efficient since the optimal interpolation kernels $\tilde{v}_1(\boldsymbol{t}),\ldots,\tilde{v}_K(\boldsymbol{t})$ are generally not compactly supported. This means that in each of the single-channel recoveries, every pixel depends on all pixels of the corresponding low-resolution image. In the next sections, we provide several efficient implementations for the multichannel hybrid Wiener filter.

IV. Efficient Implementation Schemes

We now specialize the hybrid Wiener filter to the SR setting and provide efficient implementation schemes.

We begin by noting that for any given sequence $c[\boldsymbol{n}]$, $\boldsymbol{n}\in\mathbb{Z}^d$, and function $h(\boldsymbol{t})$, $\boldsymbol{t}\in\mathbb{R}^d$, we have the CSFT relation

$$\sum_{\boldsymbol{n}\in\mathbb{Z}^d}c[\boldsymbol{n}]h(\boldsymbol{t}-\boldsymbol{n})\overset{\mathcal{F}}{\longmapsto}C(e^{j\boldsymbol{\omega}})H(\boldsymbol{\omega}), \tag{25}$$

where $C(e^{j\boldsymbol{\omega}})$ is the DSFT of $c[\boldsymbol{n}]$ and $H(\boldsymbol{\omega})$ is the CSFT of $h(\boldsymbol{t})$. Using this relation together with (24), the reconstruction formula (23) can be expressed in the frequency domain:

$$\begin{aligned}\hat{X}(\boldsymbol{\omega})&=\sum_{k=1}^{K}C_k(e^{j\boldsymbol{\omega}})\tilde{V}_k(\boldsymbol{\omega})\\&=\tilde{\boldsymbol{V}}^T(\boldsymbol{\omega})\boldsymbol{C}(e^{j\boldsymbol{\omega}})\\&=\boldsymbol{\Gamma}^T_{xy}(\boldsymbol{\omega})\left(\sum_{\boldsymbol{n}\in\mathbb{Z}}\boldsymbol{\Gamma}_{yy}(\boldsymbol{\omega}-2\pi\boldsymbol{n})\right)^{-T}\boldsymbol{C}(e^{j\boldsymbol{\omega}})\\&=\boldsymbol{\Gamma}^T_{xy}(\boldsymbol{\omega})\boldsymbol{D}(e^{j\boldsymbol{\omega}}),\end{aligned} \tag{26}$$

where we denoted $\boldsymbol{C}(\boldsymbol{\omega})=(C_1(\boldsymbol{\omega}),\ldots,C_K(\boldsymbol{\omega}))^T$ and defined

$$\boldsymbol{D}(e^{j\boldsymbol{\omega}})=\left(\sum_{\boldsymbol{n}\in\mathbb{Z}^d}\boldsymbol{\Gamma}_{yy}(\boldsymbol{\omega}-2\pi\boldsymbol{n})\right)^{-T}\boldsymbol{C}(e^{j\boldsymbol{\omega}}). \tag{27}$$

The term in the parentheses is a matrix-valued function, which is 2π-periodic in each of the elements of the frequency vector

[1]To comply with (1), the continuous-space processes $u_1(\boldsymbol{t}),\ldots,u_K(\boldsymbol{t})$ should be chosen such that, when sampled on $\mathbb{Z}^d$, produce processes whose statistics are the same as that of the discrete-space noise signals $u_1[\boldsymbol{n}],\ldots,u_K[\boldsymbol{n}]$ of (1).

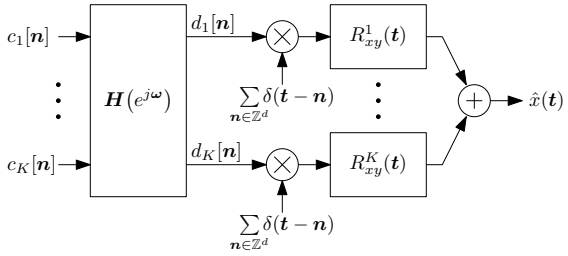


Fig. 4: Multichannel reconstruction scheme.

$\boldsymbol{\omega}$. Therefore, it corresponds to a multiple-input multiple-output (MIMO) digital filtering operation. Consequently, formula (23) can be implemented in two stages. First, the vector process $\boldsymbol{c}[\boldsymbol{n}] = (c_1[\boldsymbol{n}], \ldots, c_K[\boldsymbol{n}])^T$ is fed into the $K \times K$ MIMO digital filter

$$\boldsymbol{H}\big(e^{j\omega}\big) = \left(\sum_{\boldsymbol{n}\in\mathbb{Z}^d} \boldsymbol{\Gamma}^T_{yy}(\boldsymbol{\omega} - 2\pi\boldsymbol{n})\right)^{-1} \tag{28}$$

to obtain a "corrected" process $\boldsymbol{d}[\boldsymbol{n}] = (d_1[\boldsymbol{n}], \ldots, d_K[\boldsymbol{n}])^T$. Then, $\hat{x}(\boldsymbol{t})$ is formed using

$$\hat{x}(\boldsymbol{t}) = \sum_{k=1}^{K} \sum_{\boldsymbol{n}\in\mathbb{Z}^d} d_k[\boldsymbol{n}] R^k_{xy}(\boldsymbol{t} - \boldsymbol{n}). \tag{29}$$

This scheme is depicted in Fig. 4.

Finally, to obtain the desired discrete-space high-resolution estimate we have to convolve $\hat{x}(\boldsymbol{t})$ with the target PSF $w(\boldsymbol{t})$ and sample it on a dense grid, as implied by (17). Therefore,

$$\hat{x}_{\text{HR}}[\boldsymbol{m}] = \sum_{k=1}^{K} \sum_{\boldsymbol{n}\in\mathbb{Z}^d} d_k[\boldsymbol{n}] v_k\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{n}\Big), \tag{30}$$

with the kernels $v_k(\boldsymbol{t}) = (w * R^k_{xy})(\boldsymbol{t})$, $k = 1, \ldots, K$.

As we show next, this alternative scheme can be implemented efficiently under several conditions, which are commonly satisfied in practice. However, before discussing the implementation details, we note that the splitting of the recovery into a digital correction stage and an interpolation stage carries another important advantage. Specifically, the process of constructing the corrected images $d_1[\boldsymbol{n}], \ldots, d_K[\boldsymbol{n}]$ from the raw images $c_1[\boldsymbol{n}], \ldots, c_K[\boldsymbol{n}]$ via (27) does not depend on the magnification factor Δ. Thus, once $d_1[\boldsymbol{n}], \ldots, d_K[\boldsymbol{n}]$ are computed, we can construct $\hat{x}_{\text{HR}}[\boldsymbol{m}]$ using (30) with any desired magnification. In contrast, in most existing SR techniques, if the user wishes to change the magnification factor, then the algorithm has to be run from the beginning again.

We now proceed in making the following assumptions:

1) The noise processes $u_1(\boldsymbol{t}), \ldots, u_K(\boldsymbol{t})$ are independent of the scene $x(\boldsymbol{t})$ and are characterized by $R^{k,\ell}_{uu}(\boldsymbol{\tau}) = E[u_\ell(\boldsymbol{t})u_k(\boldsymbol{t}-\boldsymbol{\tau})] = \sigma_u^2 \delta_{k,\ell} \text{sinc}(\boldsymbol{\tau})$,
2) $\text{supp}\{s(\boldsymbol{t})\} \subseteq [-L_s, L_s]^d$,
3) $\text{supp}\{w(\boldsymbol{t})\} \subseteq [-L_w, L_w]^d$, and
4) $\text{supp}\{R_{xx}(\boldsymbol{t})\} \subseteq [-L_x, L_x]^d$.

Assumption 1 is equivalent to replacing the analog noise signals $u_1(\boldsymbol{t}), \ldots, u_K(\boldsymbol{t})$ by digital white noise processes $u_1[\boldsymbol{n}] \ldots, u_K[\boldsymbol{n}]$ with variance σ_u^2, which are added after the sampling occurs. Assumptions 2, 3 and 4 are required in order to make the recovery algorithm efficient, as we detail next.

For simplicity, we also assume in the sequel that the translations $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_K$ are in the range $[-0.5, 0.5]^d$. This does not pose any limitation since a shift $|\boldsymbol{t}_k| > 0.5$ can always be compensated for by introducing an integer shift of $-\lfloor \boldsymbol{t}_k + 0.5 \rfloor$ samples to $c_k[\boldsymbol{n}]$ prior to reconstruction.

A. Interpolation Stage

We first examine the interpolation stage, namely the computation of $\hat{x}_{\text{HR}}[\boldsymbol{m}]$ via (30). Using Assumption 1, the reconstruction kernels $R^1_{xy}(\boldsymbol{t}), \ldots, R^K_{xy}(\boldsymbol{t})$ appearing in Fig. 4 are given by

$$\begin{aligned} R^k_{xy}(\boldsymbol{t}) &= E[x(\boldsymbol{t}) y_k(\boldsymbol{0})] \\ &= E\left[x(\boldsymbol{t}) \left(\int_{\mathbb{R}^d} s(-\boldsymbol{\tau}) x(\boldsymbol{\tau} - \boldsymbol{t}_k) d\boldsymbol{\tau} + u_k[\boldsymbol{0}]\right)\right] \\ &= \int_{\mathbb{R}^d} \bar{s}(\boldsymbol{\tau}) R_{xx}(\boldsymbol{t} + \boldsymbol{t}_k - \boldsymbol{\tau}) d\boldsymbol{\tau} \\ &= (\bar{s} * R_{xx})(\boldsymbol{t} + \boldsymbol{t}_k), \end{aligned} \tag{31}$$

where we denoted $\bar{s}(\boldsymbol{t}) = s(-\boldsymbol{t})$. Consequently, the kernels $v_1(\boldsymbol{t}), \ldots, v_K(\boldsymbol{t})$ in (30), which are used to reconstruct $\hat{x}_{\text{HR}}[\boldsymbol{m}]$, are given by

$$v_k(\boldsymbol{t}) = (w * R^k_{xy})(\boldsymbol{t}) = (w * \bar{s} * R_{xx})(\boldsymbol{t} + \boldsymbol{t}_k). \tag{32}$$

Assumptions 2, 3 and 4 imply that the supports of the kernels $v_1(\boldsymbol{t}), \ldots, v_K(\boldsymbol{t})$ are finite, making the interpolation practical. Specifically, for any $\boldsymbol{m}_0 \in \mathbb{Z}^d$, the computation of $\hat{x}_{\text{HR}}[\boldsymbol{m}_0]$ involves at most $\lfloor 2(L_w + L_s + L_x) \rfloor^d$ multiplications in processing each of the corrected channels $d_1[\boldsymbol{n}], \ldots, d_K[\boldsymbol{n}]$. This sums up to a total of $K\lfloor 2(L_w + L_s + L_x) \rfloor^d$ multiplications per high-resolution pixel $\hat{x}_{\text{HR}}[\boldsymbol{m}_0]$.

This reduction in computational load in the interpolation stage comes at the cost of the MIMO digital filtering stage. Specifically, the optimal interpolation kernels $\tilde{v}_1(\boldsymbol{t}), \ldots, \tilde{v}_k(\boldsymbol{t})$ in the naive implementation (23) are generally not compactly supported. Therefore, the emergence of compactly supported kernels $v_1(\boldsymbol{t}), \ldots, v_k(\boldsymbol{t})$ in our alternative scheme means that the MIMO digital pre-filter $\boldsymbol{H}(e^{j\boldsymbol{\omega}})$ must have an infinite impulse response (IIR). However, as we show next, Assumptions 2 and 4 can be employed to implement the digital filtering stage efficiently as well.

V. Efficient Digital Pre-Filtering

We now present three methods for efficiently applying the IIR MIMO filter $\boldsymbol{H}(e^{j\boldsymbol{\omega}})$ on the samples $\boldsymbol{c}[\boldsymbol{n}]$ to obtain the corrected low-resolution signals $\boldsymbol{d}[\boldsymbol{n}]$, which are used in the interpolation stage. The first approach we study is extremely efficient, but is valid only for 1D SR applications. The other two methods are iterative, however they can be applied in arbitrary dimensions. In particular, they are applicable to spatial and spatiotemporal SR scenarios.

A. Digital Correction Stage for 1D Signals

We begin by addressing the case of uni-dimensional signals (*i.e.,* $d = 1$). In this case $x(t)$ is a signal defined over $t \in \mathbb{R}$ and the low resolution measurements $c_k[n]$ are sequences defined over $n \in \mathbb{Z}$.

A close inspection of (28), reveals that $\boldsymbol{H}(e^{j\omega})$ corresponds to the convolutional inverse of the matrix sequence $\boldsymbol{Q}[n]$, whose (k,ℓ) entry is $R_{yy}^{\ell,k}(n)$. In our setting,

$$\begin{aligned}\boldsymbol{Q}_{k,\ell}[n] &= R_{yy}^{\ell,k}(n)\\ &= E[y_k(n)y_\ell(0)]\\ &= E\left[\left(\int_{\mathbb{R}^d} s(n-\eta)x(\eta-t_k)d\eta + u_k[n]\right)\right.\\ &\qquad\left.\times\left(\int_{\mathbb{R}^d} s(-\tau)x(\tau-t_\ell)d\tau + u_\ell[0]\right)\right]\\ &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} s(n-\eta)s(-\tau)R_{xx}(\eta-\tau+t_\ell-t_k)d\tau d\eta\\ &\qquad + \sigma_u^2\delta_{k,\ell}\delta[n]\\ &= (\bar{s}*s*R_{xx})(n+t_\ell-t_k) + \sigma_u^2\delta_{k,\ell}\delta[n]. \end{aligned} \tag{33}$$

Assumptions 2 and 4 imply that the sequence $\boldsymbol{Q}[n]$ is compactly supported. This, in turn, means that the Z-transform of $\boldsymbol{Q}[n]$ can be written as

$$\boldsymbol{Q}(z) = \boldsymbol{A}_p^T z^{-p} + \ldots + \boldsymbol{A}_1^T z^{-1} + \boldsymbol{A}_0 + \boldsymbol{A}_1 z + \ldots + \boldsymbol{A}_p z^p, \tag{34}$$

where $p = \lceil 2L_s + L_x \rceil$.

Our key observation is that if $\boldsymbol{Q}(z)$ is positive definite on the unit circle $|z| = 1$, then it can be factored as

$$\boldsymbol{Q}(z) = \boldsymbol{B}(z)\boldsymbol{B}^T(z^{-1}), \tag{35}$$

where

$$\boldsymbol{B}(z) = \boldsymbol{B}_0 + \boldsymbol{B}_1 z + \ldots + \boldsymbol{B}_p z^p \tag{36}$$

is a matrix whose determinant does not vanish inside the unit circle $|z| \leq 1$. Such a factorization can be obtained *e.g.,* by any one of the methods surveyed in [23]. Using (35), we can now carry out the filtering by $\boldsymbol{H}(z) = \boldsymbol{Q}^{-1}(z)$ of (28) in two stages. Specifically, the "corrected" vector sequence $\boldsymbol{d}[n]$ is given in the frequency domain by

$$\begin{aligned}\boldsymbol{D}(z) &= \boldsymbol{H}(z)\boldsymbol{C}(z)\\ &= \boldsymbol{Q}^{-1}(z)\boldsymbol{C}(z)\\ &= \boldsymbol{B}^{-T}(z^{-1})\boldsymbol{B}^{-1}(z)\boldsymbol{C}(z). \end{aligned} \tag{37}$$

Thus, we first form an auxiliary vector sequence $\boldsymbol{c}'[n]$, by applying the MIMO filter $\boldsymbol{B}^{-1}(z)$ on the samples $\boldsymbol{c}[n]$. Then, we obtain $\boldsymbol{d}[n]$ by passing $\boldsymbol{c}'[n]$ through the MIMO filter $\boldsymbol{B}^{-T}(z^{-1})$.

The input-output relation of the first filtering operation is given by

$$\begin{aligned}\boldsymbol{C}(z) &= \boldsymbol{B}(z)\boldsymbol{C}'(z)\\ &= \left(\boldsymbol{B}_0 + \boldsymbol{B}_1 z + \ldots + \boldsymbol{B}_p z^p\right)\boldsymbol{C}'(z), \end{aligned} \tag{38}$$

which shows that

$$\boldsymbol{C}'(z) = \boldsymbol{B}_0^{-1}\left(\boldsymbol{C}(z) - \left(\boldsymbol{B}_1 z + \ldots + \boldsymbol{B}_p z^p\right)\boldsymbol{C}'(z)\right). \tag{39}$$

In the time domain, this filter can be implemented by the recursive formula

$$\boldsymbol{c}'[n] = \boldsymbol{B}_0^{-1}\left(\boldsymbol{c}[n] - \boldsymbol{B}_1\boldsymbol{c}'[n+1] - \ldots - \boldsymbol{B}_p\boldsymbol{c}'[n+p]\right) \tag{40}$$

running from right to left. Similarly, the input-output relation of the second filtering operation is given by

$$\begin{aligned}\boldsymbol{C}'(z) &= \boldsymbol{B}^T(z^{-1})\boldsymbol{D}(z)\\ &= \left(\boldsymbol{B}_0^T + \boldsymbol{B}_1^T z^{-1} + \ldots + \boldsymbol{B}_p^T z^{-p}\right)\boldsymbol{D}(z), \end{aligned} \tag{41}$$

which shows that

$$\boldsymbol{D}(z) = \boldsymbol{B}_0^{-T}\left(\boldsymbol{C}'(z) - \left(\boldsymbol{B}_1^T z^{-1} + \ldots + \boldsymbol{B}_p^T z^{-p}\right)\boldsymbol{D}(z)\right). \tag{42}$$

In the time domain, this filter can be implemented by the recursive formula

$$\boldsymbol{d}[n] = \boldsymbol{B}_0^{-T}\left(\boldsymbol{c}'[n] - \boldsymbol{B}_1^T\boldsymbol{d}[n-1] - \ldots - \boldsymbol{B}_p^T\boldsymbol{d}[n-p]\right) \tag{43}$$

running from left to right.

Due to the spectral factorization properties, the determinant of the anti-causal filter $\boldsymbol{B}^{-1}(z)$ has no poles inside the complex unit circle $|z| \leq 1$ and therefore (40) is a stable operation. Similarly, the determinant of the causal filter $\boldsymbol{B}^{-T}(z^{-1})$ has no poles outside the unit circle, implying that the recursive formula (43) is also stable. Algorithm 1 summarizes the proposed 1D SR interpolation scheme.

The technique outlined above follows similar ideas as those in the direct B-spline transform introduced in [24], which is used for single-channel spline interpolation from uniformly spaced samples. There are, however, several major differences a practitioner must be aware of, which are caused by the fact that the SR setting is more complicated. First, in contrast to the scalar interpolation scenario, here the order in which (40) and (43) are performed is important. Second, this scheme cannot be extended to multiple dimensions by operating along each dimension separately, as done in the scalar case. Specifically, even if the PSF $s(\boldsymbol{t})$ and the autocorrelation $R_{xx}(\boldsymbol{t})$ are separable functions of the coordinates $\boldsymbol{t}$, the digital filtering by $\boldsymbol{H}(e^{j\boldsymbol{\omega}})$ of (28) is generally not equivalent to applying (40) and (43) sequentially on each dimension. Last, we note that whereas in scalar interpolation the filter coefficients are computed in advance, in typical SR applications the translations $t_1, \ldots, t_K$ are estimated from the data and thus the matrices $\boldsymbol{B}_0, \ldots, \boldsymbol{B}_p$ are not known beforehand. This may complicate potential hardware implementations of the filtering stage.

B. Digital Correction Stage via the Neumann Series

Next, we propose an efficient iterative technique for implementing the filtering by $\boldsymbol{H}(e^{j\boldsymbol{\omega}})$ of (28) in arbitrary dimensions.

We begin by choosing a constant $\alpha > 0$ such that

$$\left\|\boldsymbol{I} - \alpha\boldsymbol{Q}(e^{j\boldsymbol{\omega}})\right\| < 1, \quad \forall\boldsymbol{\omega} \in \mathbb{R}^d, \tag{44}$$

where $\boldsymbol{Q}(e^{j\boldsymbol{\omega}}) = \boldsymbol{H}^{-1}(e^{j\boldsymbol{\omega}})$, $\boldsymbol{I}$ is the identity matrix and $\|\boldsymbol{A}\|$ denotes the spectral norm of a matrix $\boldsymbol{A}$. Below, we discuss how such a constant can be chosen. Now, the filtering

Algorithm 1 Fast 1D hybrid Wiener super-resolution.

Input: Low resolution sequences $c_1[n],\ldots,c_K[n]$, $n\in\mathbb{Z}$, noise variance σ_u^2, shifts $t_1,\ldots,t_K$ in the range $[-0.5,0.5]$, imaging PSF $s(t)$ supported on $[-L_s,L_s]$, target PSF $w(t)$ supported on $[-L_w,L_w]$, signal's autocorrelation $R_{xx}(t)$ supported on $[-L_x,L_x]$, magnification factor Δ.

Output: High resolution recovery $\hat{x}_{\rm HR}[n]$, $n\in\mathbb{Z}$.

1: **PSD computation:** Set $(\boldsymbol{A}_0)_{k,\ell}=(\bar{s}*s*R_{xx})(t_\ell-t_k)+\sigma_u^2\delta_{k,\ell}$ and $(\boldsymbol{A}_n)_{k,\ell}=(\bar{s}*s*R_{xx})(n+t_\ell-t_k)$, $n=1,\ldots,p$, where $p=\lceil 2L_s+L_x\rceil$.

2: **Spectral factorization:** Given the matrices $\boldsymbol{A}_0,\ldots,\boldsymbol{A}_p$, compute matrices $\boldsymbol{B}_0,\ldots,\boldsymbol{B}_p$ satisfying (35) using any matrix spectral factorization algorithm, *e.g.,* one of the methods surveyed in [23].

3: **Anti-causal MIMO filtering:** Apply (40) on the vector process $\boldsymbol{c}[n]=(c_1[n],\ldots,c_K[n])^T$ to obtain an auxiliary vector process $\boldsymbol{c}'[n]=(c_1'[n],\ldots,c_K'[n])^T$.

4: **Causal filtering:** Apply (43) on $\boldsymbol{c}'[n]$ to obtain the vector process $\boldsymbol{d}[n]=(d_1[n],\ldots,d_K[n])^T$.

5: **Interpolation:** Compute $\hat{x}_{\rm HR}[m]=\sum_{k=1}^K\sum_{n\in\mathbb{Z}}d_k[n]v_k(m/\Delta-n)$, where $v_k(t)=(w*\bar{s}*R_{xx})(t+t_k)$, $k=1,\ldots,K$.

operation (27) with $\boldsymbol{H}(e^{j\omega})$ of (28) can be expressed via the Newmann series [25]

$$\begin{aligned}\boldsymbol{D}(e^{j\omega})&=\boldsymbol{H}(e^{j\boldsymbol{\omega}})\boldsymbol{C}(e^{j\boldsymbol{\omega}})\\&=\boldsymbol{Q}^{-1}(e^{j\boldsymbol{\omega}})\boldsymbol{C}(e^{j\boldsymbol{\omega}})\\&=\alpha\left(\boldsymbol{I}-\left(\boldsymbol{I}-\alpha\boldsymbol{Q}(e^{j\boldsymbol{\omega}})\right)\right)^{-1}\boldsymbol{C}(e^{j\boldsymbol{\omega}})\\&=\alpha\sum_{r=0}^{\infty}\left(\boldsymbol{I}-\alpha\boldsymbol{Q}(e^{j\boldsymbol{\omega}})\right)^r\boldsymbol{C}(e^{j\boldsymbol{\omega}}).\end{aligned}\tag{45}$$

As we have seen in Section V-A, Assumptions 2 and 4 imply that $\boldsymbol{Q}(e^{j\boldsymbol{\omega}})$ is a finite impulse response (FIR) filter. Consequently, $\boldsymbol{I}-\alpha\boldsymbol{Q}(e^{j\boldsymbol{\omega}})$ is also an FIR filter. This shows that filtering the samples $\boldsymbol{c}[\boldsymbol{n}]$ with the IIR filter $\boldsymbol{H}(e^{j\boldsymbol{\omega}})$ can be performed by a sequence of FIR filtering operations.

To obtain a simple recursive implementation, we let

$$\hat{\boldsymbol{D}}_m(e^{j\boldsymbol{\omega}})=\alpha\sum_{r=0}^{m}\left(\boldsymbol{I}-\alpha\boldsymbol{Q}(e^{j\boldsymbol{\omega}})\right)^r\boldsymbol{C}(e^{j\boldsymbol{\omega}})\tag{46}$$

denote the approximation of $\boldsymbol{D}(e^{j\boldsymbol{\omega}})$ by the first $m+1$ terms of the Neumann series. Then $\hat{\boldsymbol{D}}_{m+1}(e^{j\boldsymbol{\omega}})$ can be calculated from $\hat{\boldsymbol{D}}_m(e^{j\boldsymbol{\omega}})$ as follows:

$$\begin{aligned}\hat{\boldsymbol{D}}_{m+1}(e^{j\boldsymbol{\omega}})&=\alpha\sum_{r=0}^{m+1}\left(\boldsymbol{I}-\alpha\boldsymbol{Q}(e^{j\boldsymbol{\omega}})\right)^r\boldsymbol{C}(e^{j\boldsymbol{\omega}})\\&=\alpha\left(\boldsymbol{C}(e^{j\boldsymbol{\omega}})+\sum_{r=1}^{m+1}\left(\boldsymbol{I}-\alpha\boldsymbol{Q}(e^{j\boldsymbol{\omega}})\right)^r\boldsymbol{C}(e^{j\boldsymbol{\omega}})\right)\\&=\alpha\left(\boldsymbol{C}(e^{j\boldsymbol{\omega}})+\left(\boldsymbol{I}-\alpha\boldsymbol{Q}(e^{j\boldsymbol{\omega}})\right)\frac{1}{\alpha}\hat{\boldsymbol{D}}_m(e^{j\boldsymbol{\omega}})\right)\\&=\hat{\boldsymbol{D}}_m(e^{j\boldsymbol{\omega}})+\alpha\left(\boldsymbol{C}(e^{j\boldsymbol{\omega}})-\boldsymbol{Q}(e^{j\boldsymbol{\omega}})\hat{\boldsymbol{D}}_m(e^{j\boldsymbol{\omega}})\right).\end{aligned}\tag{47}$$

As $\hat{\boldsymbol{D}}_m(e^{j\boldsymbol{\omega}})$ approaches $\boldsymbol{Q}^{-1}(e^{j\boldsymbol{\omega}})\boldsymbol{C}(e^{j\boldsymbol{\omega}})$, the increment $\boldsymbol{C}(e^{j\boldsymbol{\omega}})-\boldsymbol{Q}(e^{j\boldsymbol{\omega}})\hat{\boldsymbol{D}}_m(e^{j\boldsymbol{\omega}})$ approaches zero. In practice, when this increment is sufficiently small, the iterations may be terminated.

To choose a step size $\alpha>0$ that guarantees convergence, we note that the matrix $\boldsymbol{Q}(e^{j\boldsymbol{\omega}})$ is positive semi-definite. Consequently,

$$\begin{aligned}&\left\|\boldsymbol{I}-\alpha\boldsymbol{Q}(e^{j\boldsymbol{\omega}})\right\|=\\&\quad=\max\left\{1-\alpha\lambda_{\min}\left(\boldsymbol{Q}(e^{j\boldsymbol{\omega}})\right),\,\alpha\lambda_{\max}\left(\boldsymbol{Q}(e^{j\boldsymbol{\omega}})\right)-1\right\},\end{aligned}\tag{48}$$

where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote respectively the minimal and maximal eigenvalues of their matrix argument. Therefore, for (44) to hold, we need that

$$1-\alpha\lambda_{\min}\left(\boldsymbol{Q}(e^{j\boldsymbol{\omega}})\right)<1,\tag{49}$$

$$\alpha\lambda_{\max}\left(\boldsymbol{Q}(e^{j\boldsymbol{\omega}})\right)-1<1,\tag{50}$$

for all $\boldsymbol{\omega}\in[-\pi,\pi]^d$. Condition (50) is satisfied if

$$\alpha<\frac{2}{\lambda_{\max}(\boldsymbol{Q}(e^{j\boldsymbol{\omega}}))},\quad\forall\boldsymbol{\omega}\in[-\pi,\pi]^d.\tag{51}$$

Condition (49) can be fulfilled only if $\lambda_{\min}(\boldsymbol{Q}(e^{j\boldsymbol{\omega}}))>0$ for every $\boldsymbol{\omega}\in[-\pi,\pi]^d$. In this case, it is satisfied if $\alpha>0$.

In practice, it may be time-consuming to seek the worst-case $\boldsymbol{\omega}\in[-\pi,\pi]^d$ in (51) such that convergence is ensured for all frequencies. However, recall from (28) that $\boldsymbol{Q}(e^{j\boldsymbol{\omega}})$ corresponds to a sum of shifted replicas of the measurement spectrum $\boldsymbol{\Gamma}_{yy}(\boldsymbol{\omega})$. In typical SR applications, this spectrum rapidly decays as a function of frequency. Thus, it is reasonable to expect that $\lambda_{\max}(\boldsymbol{Q}(e^{j\boldsymbol{\omega}}))$ attains its maximum at $\boldsymbol{\omega}=\boldsymbol{0}$, so that (51) can be verified only there. Below, we also discuss how the step-size α can be varied from one iteration to the next, in order to achieve a faster convergence. Algorithm 2 summarizes the proposed SR interpolation scheme.

C. Digital Correction Stage via Steepest-Descent Iterations

Algorithm 2 is very efficient, since every iteration consists only of applying a MIMO FIR filter with $(2p+1)^d$ coefficients. Its disadvantage, though, is that it is commonly impossible to choose a step size α for which the convergence at all frequencies is fast. We therefore now propose a method in which the step size is updated from one iteration to the next. As we will see, this makes each iteration slower, but guarantees a rapid convergence at all frequencies.

As we have seen, the corrected signal $\boldsymbol{d}[\boldsymbol{n}]$ is obtained by applying the filter $\boldsymbol{H}(e^{j\boldsymbol{\omega}})=\boldsymbol{Q}^{-1}(e^{j\boldsymbol{\omega}})$ on the samples $\boldsymbol{c}[\boldsymbol{n}]$.

Algorithm 2 Hybrid Wiener super-resolution via Neumann iterations.

Input: Low resolution signals $c_1[\boldsymbol{n}], \ldots, c_K[\boldsymbol{n}]$, $\boldsymbol{n} \in \mathbb{Z}^d$, noise variance σ_u^2, shifts $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_K$ in the range $[-0.5, 0.5]^d$, imaging PSF $s(\boldsymbol{t})$ supported on $[-L_s, L_s]^d$, target PSF $w(\boldsymbol{t})$ supported on $[-L_w, L_w]^d$, signal's autocorrelation $R_{xx}(\boldsymbol{t})$ supported on $[-L_x, L_x]^d$, magnification factor Δ.

Output: High resolution recovery $\hat{x}_{\text{HR}}[\boldsymbol{n}]$, $\boldsymbol{n} \in \mathbb{Z}^d$.

1: **PSD computation:** Set $p = \lceil 2L_s + L_x \rceil$ and calculate the matrices $(\boldsymbol{Q}[\boldsymbol{n}])_{k,\ell} = (\bar{s} * s * R_{xx})(\boldsymbol{n} + \boldsymbol{t}_\ell - \boldsymbol{t}_k) + \sigma_u^2 \delta_{k,\ell} \delta[\boldsymbol{n}]$ for $\boldsymbol{n} \in [-p, p]^d$.
2: **Step size calculation:** Choose a step size α satisfying $0 < \alpha < 2/\lambda_{\max}(\sum_{\boldsymbol{n} \in [-p,p]^d} \boldsymbol{Q}[\boldsymbol{n}])$.
3: **Initialization:** Set $m = 0$, $\boldsymbol{d}_m[\boldsymbol{n}] = \boldsymbol{0}$, for all $\boldsymbol{n} \in \mathbb{Z}^d$.
4: **repeat**
5: Compute $\boldsymbol{\varepsilon}_m[\boldsymbol{\ell}] = \boldsymbol{c}[\boldsymbol{\ell}] - \sum_{\boldsymbol{r} \in [-p,p]^d} \boldsymbol{Q}[\boldsymbol{r}] \boldsymbol{d}_m[\boldsymbol{\ell} - \boldsymbol{r}]$ for every $\boldsymbol{\ell} \in \mathbb{Z}^d$.
6: To use the optimal step-size, compute $\tilde{\boldsymbol{\varepsilon}}_m[\boldsymbol{\ell}] = \sum_{\boldsymbol{r} \in [-p,p]^d} \boldsymbol{Q}[\boldsymbol{r}] \boldsymbol{\varepsilon}_m[\boldsymbol{\ell} - \boldsymbol{r}]$ for every $\boldsymbol{\ell} \in \mathbb{Z}^d$ and set $\alpha_m = (\sum_{\boldsymbol{n} \in \mathbb{Z}^d} \boldsymbol{\varepsilon}_m^T[\boldsymbol{n}] \tilde{\boldsymbol{\varepsilon}}_m[\boldsymbol{n}]) / (\sum_{\boldsymbol{n} \in \mathbb{Z}^d} \|\tilde{\boldsymbol{\varepsilon}}_m[\boldsymbol{n}]\|^2)$. Otherwise set $\alpha_m = \alpha$.
7: Set $\boldsymbol{d}_{m+1}[\boldsymbol{n}] = \boldsymbol{d}_m[\boldsymbol{n}] + \alpha_m \boldsymbol{\varepsilon}_m[\boldsymbol{n}]$ for every $\boldsymbol{n} \in \mathbb{Z}^d$.
8: Set $m \leftarrow m + 1$.
9: **until** $\sum_{\boldsymbol{n} \in \mathbb{Z}^d} \|\boldsymbol{\varepsilon}_m[\boldsymbol{n}]\|^2$ ceases to decrease.
10: **Interpolation:** Compute $\hat{x}_{\text{HR}}[\boldsymbol{\ell}] = \sum_{k=1}^K \sum_{\boldsymbol{n} \in \mathbb{Z}} d_k[\boldsymbol{n}] v_k(\boldsymbol{\ell}/\Delta - \boldsymbol{n})$, where $v_k(\boldsymbol{t}) = (w * \bar{s} * R_{xx})(\boldsymbol{t} + \boldsymbol{t}_k)$, $k = 1, \ldots, K$.

Therefore, for every $\boldsymbol{\omega} \in [-\pi, \pi]^d$, the vector $\boldsymbol{D}(e^{j\boldsymbol{\omega}})$ is the minimizer of the objective

$$f(\boldsymbol{D}(e^{j\boldsymbol{\omega}})) = \left\| \boldsymbol{Q}(e^{j\boldsymbol{\omega}}) \boldsymbol{D}(e^{j\boldsymbol{\omega}}) - \boldsymbol{C}(e^{j\boldsymbol{\omega}}) \right\|^2. \tag{52}$$

The gradient of $f(\boldsymbol{D}(e^{j\omega}))$ with respect to $\boldsymbol{D}(e^{j\boldsymbol{\omega}})$ is given by

$$\nabla_{\boldsymbol{D}} f(e^{j\omega}) = 2\boldsymbol{Q}^H(e^{j\boldsymbol{\omega}}) \left(\boldsymbol{Q}(e^{j\boldsymbol{\omega}}) \boldsymbol{D}(e^{j\boldsymbol{\omega}}) - \boldsymbol{C}(e^{j\boldsymbol{\omega}}) \right). \tag{53}$$

Noting that in our setting $\boldsymbol{Q}^H(e^{j\boldsymbol{\omega}}) = \boldsymbol{Q}(e^{j\boldsymbol{\omega}})$, the minimizer of $f(e^{j\boldsymbol{\omega}})$ can be found using the steepest descent iterations

$$\hat{\boldsymbol{D}}_{m+1}(e^{j\boldsymbol{\omega}}) = \hat{\boldsymbol{D}}_m(e^{j\boldsymbol{\omega}}) + \alpha_m \boldsymbol{Q}(e^{j\boldsymbol{\omega}}) \left(\boldsymbol{C}(e^{j\boldsymbol{\omega}}) - \boldsymbol{Q}(e^{j\boldsymbol{\omega}}) \hat{\boldsymbol{D}}_m(e^{j\boldsymbol{\omega}}) \right) \tag{54}$$

Note that (54) is the same as the Neumann update rule (47), besides the factor $\boldsymbol{Q}(e^{j\boldsymbol{\omega}})$ and the fact that the step-size α_m may change from one iteration to the next. If α_m is properly chosen at each iteration, then the sequence of approximations $\hat{\boldsymbol{D}}_m(e^{j\boldsymbol{\omega}})$ will converge to $\boldsymbol{D}(e^{j\boldsymbol{\omega}})$ [26].

We would like to choose a step size α_m, such that the energy of the error is minimized. Specifically, we would like to minimize

$$\int_{\boldsymbol{\omega} \in [-\pi,\pi]^d} f\left(\hat{\boldsymbol{D}}_{m+1}(e^{j\boldsymbol{\omega}}) \right) d\boldsymbol{\omega} = \int_{\boldsymbol{\omega} \in [-\pi,\pi]^d} f\Big(\hat{\boldsymbol{D}}_m(e^{j\boldsymbol{\omega}}) + \alpha_m \boldsymbol{Q}(e^{j\boldsymbol{\omega}}) \left(\boldsymbol{C}(e^{j\boldsymbol{\omega}}) - \boldsymbol{Q}(e^{j\boldsymbol{\omega}}) \hat{\boldsymbol{D}}_m(e^{j\boldsymbol{\omega}}) \right) \Big) d\boldsymbol{\omega}. \tag{55}$$

Substituting the expression for $f(\boldsymbol{D}(e^{j\omega}))$ from (52) and equating the derivative with respect to α_m to zero, we find that the optimal step size is given by

$$\alpha_m = \frac{\int_{\boldsymbol{\omega} \in [-\pi,\pi]^d} \|\tilde{\boldsymbol{E}}_m(e^{j\boldsymbol{\omega}})\|^2 d\boldsymbol{\omega}}{\int_{\boldsymbol{\omega} \in [-\pi,\pi]^d} \|\boldsymbol{Q}(e^{j\boldsymbol{\omega}}) \tilde{\boldsymbol{E}}_m(e^{j\boldsymbol{\omega}})\|^2 d\boldsymbol{\omega}} = \frac{\sum_{\boldsymbol{n} \in \mathbb{Z}^d} \|\tilde{\boldsymbol{\varepsilon}}_m[\boldsymbol{n}]\|^2}{\sum_{\boldsymbol{n} \in \mathbb{Z}^d} \left\| \sum_{\boldsymbol{\ell} \in [-p,p]^d} \boldsymbol{Q}[\boldsymbol{\ell}] \tilde{\boldsymbol{\varepsilon}}_m[\boldsymbol{n} - \boldsymbol{\ell}] \right\|^2}, \tag{56}$$

where $\tilde{\boldsymbol{E}}_m(e^{j\boldsymbol{\omega}}) = \boldsymbol{Q}(e^{j\boldsymbol{\omega}})(\boldsymbol{C}(e^{j\boldsymbol{\omega}}) - \boldsymbol{Q}(e^{j\boldsymbol{\omega}}) \hat{\boldsymbol{D}}_m(e^{j\boldsymbol{\omega}}))$ and $\tilde{\boldsymbol{\varepsilon}}_m[\boldsymbol{n}]$ is its inverse DSFT. Algorithm 3 summarizes the proposed SR interpolation scheme.

It is important to note that each steepest-descent iteration comprises three applications of the FIR MIMO filter $\boldsymbol{Q}[\boldsymbol{n}]$, as opposed to the single filtering operation required in the Neumann method. Therefore, for Algorithm 3 to be preferable to Algorithm 2 in terms of running time, it has to converge to the optimum at least three times faster. Unfortunately, this is usually not the case. In typical situations, its first few iterations lead to a dramatic increase in the error, but its convergence becomes very slow near the optimum. This phenomenon of the steepest-descent method is well known [26].

Interestingly, the computational complexity of Algorithm 3 can be reduced, while retaining the benefits of a varying step-size. To do this, we note that the direction of the Neumann update (47) is related to that of the steepest-descent (54), via multiplication by the positive-definite matrix $\boldsymbol{Q}^{-1}(e^{j\omega})$. Thus, the objective is guaranteed to decrease in the direction of the Neumann update (47). Now, we can compute the step-size α_m leading to the maximal reduction in the error for the Neumann iterations, similar to equations (55) and (56), which were relevant for the steepest-descent method. The resulting optimal Neumann step-size is given by

$$\alpha_m = \frac{\sum_{\boldsymbol{n} \in \mathbb{Z}^d} \boldsymbol{\varepsilon}_m^T[\boldsymbol{n}] \tilde{\boldsymbol{\varepsilon}}_m[\boldsymbol{n}]}{\sum_{\boldsymbol{n} \in \mathbb{Z}^d} \|\tilde{\boldsymbol{\varepsilon}}_m[\boldsymbol{n}]\|^2}, \tag{57}$$

where $\boldsymbol{\varepsilon}_m[\boldsymbol{\ell}] = \boldsymbol{c}[\boldsymbol{\ell}] - \sum_{\boldsymbol{r} \in [-p,p]^d} \boldsymbol{Q}[\boldsymbol{r}] \boldsymbol{d}_m[\boldsymbol{\ell} - \boldsymbol{r}]$, and $\tilde{\boldsymbol{\varepsilon}}_m[\boldsymbol{n}]$ is as in (56). This option of using a varying step-size in the Neumann iterations is presented in Algorithm 2. It requires

Algorithm 3 Hybrid Wiener super-resolution via steepest descent iterations.

Input: Low resolution signals $c_1[\boldsymbol{n}], \ldots, c_K[\boldsymbol{n}]$, $\boldsymbol{n} \in \mathbb{Z}^d$, noise variance σ_u^2, shifts $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_K$ in the range $[-0.5, 0.5]^d$, imaging PSF $s(\boldsymbol{t})$ supported on $[-L_s, L_s]^d$, target PSF $w(\boldsymbol{t})$ supported on $[-L_w, L_w]^d$, signal's autocorrelation $R_{xx}(\boldsymbol{t})$ supported on $[-L_x, L_x]^d$, magnification factor Δ.

Output: High resolution recovery $\hat{x}_{\text{HR}}[\boldsymbol{n}]$, $\boldsymbol{n} \in \mathbb{Z}^d$.

PSD computation: Set $p = \lceil 2L_s + L_x \rceil$ and calculate the matrices $(\boldsymbol{Q}[\boldsymbol{n}])_{k,\ell} = (\bar{s} * s * R_{xx})(\boldsymbol{n} + \boldsymbol{t}_\ell - \boldsymbol{t}_k) + \sigma_u^2 \delta_{k,\ell} \delta[\boldsymbol{n}]$ for $\boldsymbol{n} \in [-p, p]^d$.

Initialization: Set $m = 0$, $\boldsymbol{d}_m[\boldsymbol{n}] = \boldsymbol{0}$, for all $\boldsymbol{n} \in \mathbb{Z}^d$.

repeat

  Compute $\boldsymbol{\varepsilon}_m[\boldsymbol{\ell}] = \boldsymbol{c}[\boldsymbol{\ell}] - \sum_{\boldsymbol{r} \in [-p,p]^d} \boldsymbol{Q}[\boldsymbol{r}] \boldsymbol{d}_m[\boldsymbol{\ell} - \boldsymbol{r}]$ for every $\boldsymbol{\ell} \in \mathbb{Z}^d$.

  Compute $\tilde{\boldsymbol{\varepsilon}}_m[\boldsymbol{\ell}] = \sum_{\boldsymbol{r} \in [-p,p]^d} \boldsymbol{Q}[\boldsymbol{r}] \boldsymbol{\varepsilon}_m[\boldsymbol{\ell} - \boldsymbol{r}]$ for every $\boldsymbol{\ell} \in \mathbb{Z}^d$.

  Compute $\alpha_m = (\sum_{\boldsymbol{n} \in \mathbb{Z}^d} \|\tilde{\boldsymbol{\varepsilon}}_m[\boldsymbol{n}]\|^2) / (\sum_{\boldsymbol{n} \in \mathbb{Z}^d} \| \sum_{\boldsymbol{\ell} \in [-p,p]^d} \boldsymbol{Q}[\boldsymbol{\ell}] \tilde{\boldsymbol{\varepsilon}}_m[\boldsymbol{n} - \boldsymbol{\ell}] \|^2)$.

  Set $\boldsymbol{d}_{m+1}[\boldsymbol{n}] = \boldsymbol{d}_m[\boldsymbol{n}] + \alpha_m \tilde{\boldsymbol{\varepsilon}}_m[\boldsymbol{n}]$ for every $\boldsymbol{n} \in \mathbb{Z}^d$.

  Set $m \leftarrow m + 1$.

until $\sum_{\boldsymbol{n} \in \mathbb{Z}^d} \|\boldsymbol{\varepsilon}[\boldsymbol{n}]\|^2$ ceases to decrease.

Interpolation: Compute $\hat{x}_{\text{HR}}[\boldsymbol{\ell}] = \sum_{k=1}^K \sum_{\boldsymbol{n} \in \mathbb{Z}} d_k[\boldsymbol{n}] v_k(\boldsymbol{\ell}/\Delta - \boldsymbol{n})$, where $v_k(\boldsymbol{t}) = (w * R_{xy}^k)(\boldsymbol{t})$, $k = 1, \ldots, K$.

two filtering operations in each iteration, as opposed to the three required in Algorithm 3.

D. Comparison

For 1D applications, Algorithm 1 seems to be advantageous over Algorithms 2 and 3 due to its non-iterative form. Specifically, it does not suffer from error due to a finite number of iterations. Furthermore, its time-consumption is known in advance and does not depend on the underlying signal.

For SR in two or more dimensions, one must use either Algorithm 2 (with a fixed or varying step-size) or Algorithm 3. The rate of convergence of the steepest descent method (Algorithm 3) is known to be very slow near the optimum, especially whenever the ratio $\lambda_{\max}(\boldsymbol{Q}(e^{j\boldsymbol{\omega}}))/\lambda_{\min}(\boldsymbol{Q}(e^{j\boldsymbol{\omega}}))$ is large [26]. The latter is controlled by the noise variance σ_u^2, which appears in the diagonal elements of $\boldsymbol{Q}(e^{j\boldsymbol{\omega}})$. The condition number of $\boldsymbol{Q}(e^{j\boldsymbol{\omega}})$ also affects the convergence rate of the Neumann method in a similar manner.

Figure 5 depicts the root MSE (RMSE) as a function of the number of iterations in the constant and the varying step-size Neuman methods, and in the steepest descent algorithm. Figures 5a and 5b differ by the noise level σ_u^2. The former was run with a signal-to-noise ratio (SNR) of 0dB whereas the latter with 8dB. These experiments validate that the convergence rate of all algorithms is indeed higher for low SNRs. Furthermore, it is evident that the steepest-descent method has the slowest convergence rate. That, together with the fact that its iterations are more computational demanding than the other methods, make it the least favorable of all three approaches.

The varying step-size Neumann iterations provide the fastest convergence. However, the gain with respect to using a constant step-size is rather small. On the other hand, each iteration of the varying step-size Neuman approach requires twice the amount of computations as its fixed step-size counterpart. Therefore, we conclude that the best approach in terms of overall time consumption, is Algorithm 2 with a constant step-size.

VI. Error Analysis and Patterns

In [27], [28], the authors investigated bounds on the achievable MSE in SR scenarios. These analyses, however, did not incorporate any assumed prior knowledge on the underlying scene. As we show next, one of the benefits of our approach, is that we can obtain closed form expressions for the MSE in our setting. This is due to the fact that we assume knowledge of the statistics of $x(\boldsymbol{t})$. The error analysis we provide in this section is particularly useful in predicting the appearance and the extent to which periodic structures appear in the reconstructed high-resolution recovery $\hat{x}_{\text{HR}}[\boldsymbol{n}]$.

Our goal is to obtain an expression for the MSE $\epsilon^2[\boldsymbol{m}]$ defined in (16), which is the error in estimating $x_{\text{HR}}[\boldsymbol{m}]$ by $\hat{x}_{\text{HR}}[\boldsymbol{m}]$. Substituting (4) and (17), $\epsilon^2[\boldsymbol{m}]$ can be expressed as

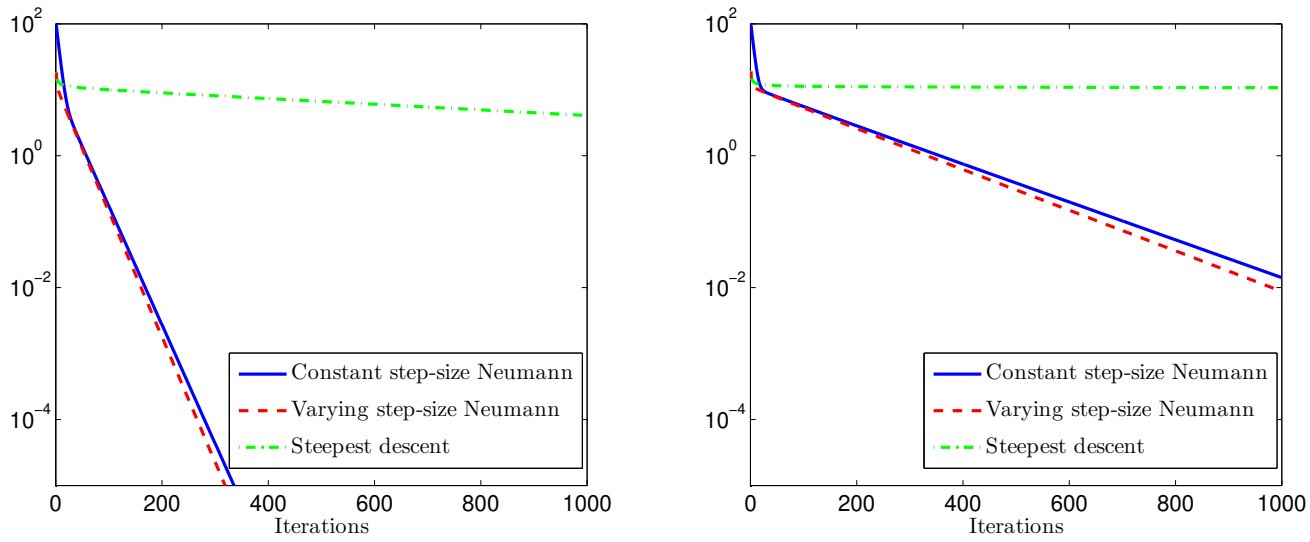
$$\begin{aligned}
\epsilon^2[\boldsymbol{m}] &= E\Big[\big(\hat{x}_{\text{HR}}[\boldsymbol{n}] - x_{\text{HR}}[\boldsymbol{n}]\big)^2\Big] \\
&= E\left[\left(\int_{\boldsymbol{t} \in \mathbb{R}^d} (x(\boldsymbol{t}) - \hat{x}(\boldsymbol{t})) w\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{t}\Big)\, d\boldsymbol{t}\right)^2\right] \\
&= E\left[\int_{\boldsymbol{t} \in \mathbb{R}^d} \int_{\boldsymbol{\eta} \in \mathbb{R}^d} (x(\boldsymbol{t}) - \hat{x}(\boldsymbol{t}))(x(\boldsymbol{\eta}) - \hat{x}(\boldsymbol{\eta})) \right. \\
&\qquad \left. \times w\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{t}\Big)\, w\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{\eta}\Big)\, d\boldsymbol{t} d\boldsymbol{\eta}\right] \\
&= \int_{\boldsymbol{t} \in \mathbb{R}^d} \int_{\boldsymbol{\eta} \in \mathbb{R}^d} e(\boldsymbol{t}, \boldsymbol{\eta}) w\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{t}\Big)\, w\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{\eta}\Big)\, d\boldsymbol{t} d\boldsymbol{\eta},
\end{aligned} \tag{58}$$

where $\hat{x}(\boldsymbol{t})$ denotes the LMMSE estimate of the continuous scene $x(\boldsymbol{t})$, and

$$e(\boldsymbol{t}, \boldsymbol{\eta}) = E[(x(\boldsymbol{t}) - \hat{x}(\boldsymbol{t}))(x(\boldsymbol{\eta}) - \hat{x}(\boldsymbol{\eta}))] \tag{59}$$

is the covariance function of the error in estimating $x(\boldsymbol{t})$ by $\hat{x}(\boldsymbol{t})$. Therefore, the discrete-space error pattern $\epsilon^2[\boldsymbol{m}]$ is associated with the continuous-space error covariance function of the vector hybrid Wiener filter.

Theorem 3 (Error of the vector hybrid Wiener filter): *The error covariance* (59) *of the LMMSE estimate* $\hat{x}(\boldsymbol{t})$ *of* $x(\boldsymbol{t})$



(a) (b)

Fig. 5: RMSE as a function of iterations. (a) SNR of 0dB. (b) SNR of 8dB.

based on the measurements $y_1(\boldsymbol{n}), \ldots, y_K(\boldsymbol{n})$, $\boldsymbol{n} \in \mathbb{Z}^d$, *is given by*

$$e(\boldsymbol{t}, \boldsymbol{\eta}) = R_{xx}(\boldsymbol{t}-\boldsymbol{\eta}) - \sum_{k=1}^{K} \sum_{\boldsymbol{n}\in\mathbb{Z}^d} \tilde{v}_k(\boldsymbol{t}-\boldsymbol{n}) R_{xy}^k(\boldsymbol{\eta}-\boldsymbol{n}), \quad (60)$$

where $\tilde{v}_1(\boldsymbol{t}), \ldots, \tilde{v}_K(\boldsymbol{t})$ *are given by* (24).

Proof: See Appendix B. ∎

Substituting (60) into (58), we have

$$\begin{aligned}
\epsilon^2[\boldsymbol{m}] &= \\
&= \int_{\boldsymbol{t}\in\mathbb{R}^d} \int_{\boldsymbol{\eta}\in\mathbb{R}^d} R_{xx}(\boldsymbol{t}-\boldsymbol{\eta}) w\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{t}\Big)\, w\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{\eta}\Big)\, d\boldsymbol{t} d\boldsymbol{\eta} \\
&\quad - \sum_{k=1}^{K} \sum_{\boldsymbol{n}\in\mathbb{Z}^d} \left(\int_{\boldsymbol{t}\in\mathbb{R}^d} \tilde{v}_k(\boldsymbol{t}-\boldsymbol{n}) w\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{t}\Big)\, d\boldsymbol{t} \right) \\
&\qquad \times \left(\int_{\boldsymbol{t}\in\mathbb{R}^d} R_{xy}^k(\boldsymbol{\eta}-\boldsymbol{n}) w\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{\eta}\Big)\, d\boldsymbol{\eta} \right) \\
&= \int_{\boldsymbol{\eta}\in\mathbb{R}^d} (R_{xx} * w)\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{\eta}\Big)\, w\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{\eta}\Big)\, d\boldsymbol{\eta} \\
&\quad - \sum_{k=1}^{K} \sum_{\boldsymbol{n}\in\mathbb{Z}^d} (\tilde{v}_k * w)\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{n}\Big)\, (R_{xy}^k * w)\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{n}\Big).
\end{aligned} \quad (61)$$

With a change of variables $\boldsymbol{\eta} \leftarrow \boldsymbol{m}/\Delta - \boldsymbol{\eta}$, and using the notation $v_k(\boldsymbol{t}) = (R_{xy}^k * w)(\boldsymbol{t})$ introduced in (30), this expression simplifies to

$$\begin{aligned}
\epsilon^2[\boldsymbol{m}] &= \int_{\boldsymbol{\eta}\in\mathbb{R}^d} (R_{xx} * w)(\boldsymbol{\eta}) w(\boldsymbol{\eta}) d\boldsymbol{\eta} \\
&\quad - \sum_{k=1}^{K} \sum_{\boldsymbol{n}\in\mathbb{Z}^d} (\tilde{v}_k * w)\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{n}\Big)\, v_k\Big(\frac{\boldsymbol{m}}{\Delta} - \boldsymbol{n}\Big).
\end{aligned} \quad (62)$$

The first term in this expression is constant. The second term, however, is a function of $\boldsymbol{m}$. It can be seen that if the magnification factor Δ is an integer, then the error pattern $\epsilon^2[\boldsymbol{m}]$ is periodic with period Δ.

The above exposition is important as it reveals the fact that some pixels in $x_{\text{HR}}[\boldsymbol{m}]$ are harder to estimate than others. As a consequence, the recovery $\hat{x}_{\text{HR}}[\boldsymbol{m}]$ will typically exhibit some sort of periodic structure. This effect is very well known in single-image interpolation. The most common example is the nearest-neighbor interpolation method, which produces block-artifacts, but the effect is also apparent in bilinear interpolation, bicubic recovery, etc. [14]. As discussed in [14], the narrower the interpolation kernel is, the stronger the effect of periodic structure appearing in the restored image. In our multichannel setting, the interpolation kernels are determined by the cross-correlation functions $R_{xy}^1(\boldsymbol{t}), \ldots, R_{xy}^K(\boldsymbol{t})$, which correspond to shifted versions of $R_{xx}(\boldsymbol{t})$ when the noise is independent of the scene. Therefore, the narrower $R_{xx}(\boldsymbol{t})$ is, the stronger the periodic artifacts in $\hat{x}_{\text{HR}}[\boldsymbol{m}]$ will be.

It is important to note that the periodic structure in $\hat{x}_{\text{HR}}[\boldsymbol{m}]$ can be eliminated, for example by using bandlimited interpolation kernels [14]. However, while suppressing artifacts, this would lead to a large reconstruction error. Thus, there is a tradeoff between periodic artifacts and reconstruction error. By tuning the width of $R_{xx}(\boldsymbol{t})$ we may produce a smooth recovery which substantially deviates from $x_{\text{HR}}[\boldsymbol{m}]$, or one that contains periodic artifacts but is more loyal to $x_{\text{HR}}[\boldsymbol{m}]$. This effect will be demonstrated in Section VIII, where we present SR results on images.

VII. Spline Super-Resolution

In practical imaging scenarios, the PSD $\Gamma_{xx}(\boldsymbol{\omega})$ should be chosen to roughly match the typical frequency content of the scenes encountered in a specific application (*e.g.*, medical images, natural images, etc.). In the single-image

interpolation literature, several authors have used the Matérn class of PSD functions [15], [16], which has the form $\Gamma_{xx}(\boldsymbol{\omega}) = \alpha(\|\boldsymbol{\omega}\|^2 + \beta^2)^{-(\gamma+d/2)}$ for some constants α, β and γ. It has been demonstrated that this type of prior is suitable for natural images and leads to better reconstruction results than the commonly used bilinear interpolation, Keys' bicubic interpolation [29], and Shannon's sinc interpolation. Nevertheless, the drawback of the Matérn prior is that, as shown in [30], its associated autocorrelation function $R_{xx}(\boldsymbol{t})$ is not compactly supported. This violates our Assumption 4 in Section IV.

As an alternative class of autocorrelation functions, whose frequency content decays polynomially, we now consider B-splines. A B-spline of degree N, denoted $\beta^N(t)$ is the function obtained by the $(N+1)$-fold convolution of the unit square

$$\beta^0(\boldsymbol{t}) = \begin{cases} 1 & \boldsymbol{t} \in \left[-\frac{1}{2}, \frac{1}{2}\right]^d \\ 0 & \text{else.} \end{cases} \tag{63}$$

In single-image recovery, if both the PSF $s(\boldsymbol{t})$ and the autocorrelation $R_{xx}(\boldsymbol{t})$ are B-splines of some degree, then the (scalar) hybrid Wiener filter corresponds to the widely used spline interpolation technique [30]. Consequently, in our multi-channel setting, we refer to the recovery resulting from the vector hybrid Wiener filter as *spline super-resolution* (SSR). Since B-splines all have a bell-like shape whose width is controlled by their degree, modeling the PSF $s(\boldsymbol{t})$ as a B-spline is adequate for many imaging devices. In particular, if the blur due to the lens is negligible with respect to pixel size, then $s(\boldsymbol{t})$ can be quite accurately modeled as a B-spline of degree 0. Higher-order B-splines can be used when the lens' blur is more significant.

Assume that the imaging PSF $s(\boldsymbol{t})$ is a B-spline of degree D_s, the autocorrelation $R_{xx}(\boldsymbol{t})$ is a B-spline of degree D_x, and the target PSF is a delta function $w(\boldsymbol{t}) = \delta(\boldsymbol{t})$. Then from (32), we see that the interpolation kernels $v_1(\boldsymbol{t}), \ldots, v_K(\boldsymbol{t})$ are shifted versions of B-splines of degree $D_x + D_s + 1$. Thus, the interpolation stage in our approach reduces to performing standard spline interpolation[2], with appropriate shift, on each of the low-resolution images, and then summing the results. This can be done efficiently, by relying on the fact that d-dimensional B-spline functions are separable and compactly supported. In particular, if $D_x = D_s = 0$, then the interpolation stage comprises K bilinear interpolation procedures.

The FIR filter $\boldsymbol{Q}[\boldsymbol{n}]$ needed for the digital correction stage can also be computed simply in the SSR method. Specifically, the elements of the MIMO filter $\boldsymbol{Q}[\boldsymbol{n}]$, appearing in Algorithms 2 and 3, involve samples of the function $(\bar{s} * s * R_{xx})(\boldsymbol{t})$. In our SSR approach, this function is a B-spline of degree $2D_s + D_x + 2$, and thus possesses a closed form expression.

As discussed in Section VI and will be demonstrated in the next section, the choice of the order D_x of the autocorrelation B-spline is guided by two conflicting desires. On one hand, a small D_x corresponds to a narrow autocorrelation function $R_{xx}(\boldsymbol{t})$, which leads to a sharp recovered image, but with strong periodic artifacts. A large degree D_x corresponds to a wide $R_{xx}(\boldsymbol{t})$, leading to a smooth recovery, but with a large reconstruction error. In practice, we found that D_x should be chosen to be 1 or 2 in order to obtain satisfying results.

[2]Without the associated pre-filtering, which is used in spline interpolation.

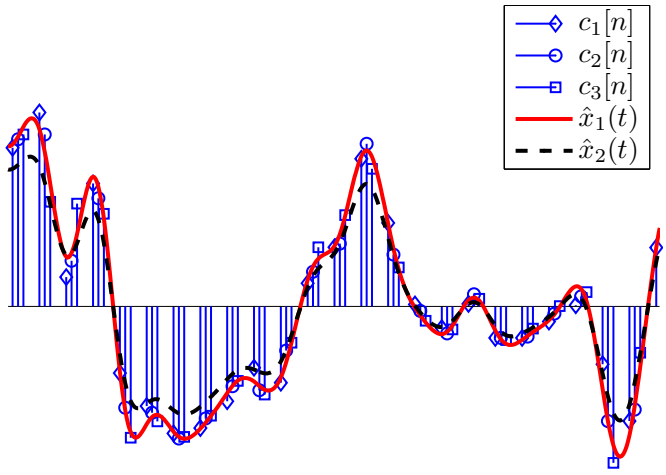


Fig. 6: 1D interpolation using the SSP algorithm.

VIII. Experiments

We now demonstrate our SR interpolation algorithm via simulations and experiments on real-world data.

A. 1D Super-Resolution

We begin by demonstrating the behavior of our SSR algorithm in the task of recovering 1D signals. Figure 6 depicts two 1D interpolation examples produced by Algorithm 1. Here, the PSF $s(t)$ is a B-spline of degree 0 (namely a rectangular window of width 1), the auto-correlation $R_{xx}(t)$ is a B-spline of degree 2, and the shifts are $(t_1, t_2, t_3) = (-0.1, 0, 0.1)$. The solid curve is the recovery $\hat{x}(t)$ obtained when setting $\sigma_u^2 = 0$. This reconstruction is consistent in the sense that if we were to pass $\hat{x}(t)$ through $s(t)$ and sample it, then the resulting samples would coincide with the measured sequences $c_1[n]$, $c_2[n]$, and $c_3[n]$.

The dashed curve is the estimate obtained when setting the noise variance σ_u^2 to be equal to the variance of the clean measurement sequences. In other words, the noise level is assumed to be 0dB. As can be seen, in this case $\hat{x}(t)$ tends to be smoother at the cost of deviating from the measured samples. This a result of the fact that the samples are assumed to be contaminated with white noise, whereas the signal is known to have little energy in the high frequencies. Therefore, in this setting the high frequencies of the LMMSE estimate are attenuated with respect to the noiseless case.

B. 2D Super-Resolution

We now present several 2D experiments with the SSR algorithm. In all these experiments, the PSF $s(\boldsymbol{t})$ was taken to be a rectangular kernel, namely a B-spline of degree $D_s = 0$. We applied the Neumann method (Algorithm 2) with a constant step-size.

We begin by visualizing the general flow of our approach. Figure 7a shows (a portion of) six low-resolution images

of a pine tree, taken with an Olympus C-60 camera. The translations $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_6$ were estimated from the images. Here, we set the autocorrelation B-spline degree to $D_x = 0$, and tuned σ_u^2 to get an assumed SNR of 0dB. In this setting, the reconstruction kernels are B-splines of degree 1, so that the recovery stage involves applying bilinear interpolation to each of the corrected images $d_1[\boldsymbol{n}], \ldots, d_6[\boldsymbol{n}]$.

A naive SR approach would be to apply bilinear interpolation to the original images $c_1[\boldsymbol{n}], \ldots, c_6[\boldsymbol{n}]$ without performing the correction stage. However, as can be seen in Fig. 7b, the corrected images $d_1[\boldsymbol{n}], \ldots, d_6[\boldsymbol{n}]$ are very different from the original ones. In particular, the correction process is highly dependent on the shifts between the images. Figures 8a and 8b depict respectively the restoration obtained with and without the correction stage. Evidently, the recovery without the pre-filter is blurred with respect to the one with the pre-filter.

Next, we compare the performance of the SSR algorithm to that of several prominent SR techniques, which rely on the discrete model (2). A common method for quantitative comparison between SR methods is to produce a set of low-resolution images $c_1[\boldsymbol{n}], \ldots, c_K[\boldsymbol{n}]$ from a known high-resolution image $\mathbf{x}_{\text{HR}}[\boldsymbol{n}]$, and then measure the MSE $E[(x_{\text{HR}}[\boldsymbol{n}] - \hat{x}_{\text{HR}}[\boldsymbol{n}])^2]$ attained by each of the studied methods. However, by doing that we would actually conform to the the incorrect discrete model (2). As we have seen, in real-world scenarios $c_1[\boldsymbol{n}], \ldots, c_K[\boldsymbol{n}]$ cannot be produced from $x_{\text{HR}}[\boldsymbol{n}]$. Therefore, in the sequel we focus on qualitative comparisons on real-world images.

The discrete SR algorithms we compare against are shift-and-add (S&A) with Wiener deconvolution [31], and the methods reported in [32], namely Median S&A with Wiener deconvolution, ℓ_1-norm regularization, ℓ_2-norm regularization, and robust recovery (ℓ_1-norm fidelity) with ℓ_1-norm regularization. In all our experiments, we supplied these algorithms with the discrete-space PSF, which is the counterpart of $s(\boldsymbol{t}) = \beta^0(\boldsymbol{t})$ used in our algorithm. These algorithms are compared with the SSR approach proposed here, with autocorrelation B-spline degrees 0, 1, and 2, which we term respectively SSR0, SSR1, and SSR2. The noise level σ_u^2 was set such that the resulting SNR was 0dB, 8dB and 10dB in the SSR0, SSR1 and SSR2 methods respectively. We used the same estimate for the translations in all algorithms.

Figure 9 shows the SR recoveries obtained from the first $K = 20$ frames of the sequence Disk taken from [33]. As can be seen, the choice of the autocorrelation B-spline degree controls the tradeoff between the recovery of fine features and artifacts. Algorithm SSR1 seems to achieve a good balance between these properties, and together with the ℓ_2-norm regularization approach, yields the best performance (note in particular the recovery of the letters 'R' 'E' and 'F'). A similar behavior can be observed in Fig. 10, where we used the 28-frame sequence Text taken from [33], and in Fig. 11, in which the 6 images from Fig. 7a were processed.

IX. Conclusion

In this paper, we analyzed the validity of the widely used discrete-space formulation of the SR problem. We showed that discrete modeling generally cannot be loyal to the actual physical setting. As a result, algorithms which do not treat the problem from a continuous-space interpolation viewpoint, inevitably suffer from a model miss-match error. This motivated us to model the scene as a continuous-space random process with known statistics, and seek its LMMSE estimate given the observed discrete-space measurements. We derived a closed form expression for the LMMSE estimator, which we coin the *vector hybrid Wiener filter*. We also provided an analytic formula for the MSE of our method, which shed light on the emergence of periodic artifacts in the recovered signal. We presented three efficient implementations of the vector hybrid Wiener filter and showed that when the signal's autocorrelation is a B-spline function, these methods constitute a natural extension of the widely used single-channel spline interpolation to the multi-channel setup. Experiments on real-world data demonstrated the effectiveness of our proposed approach.

Appendix A
Proof of Theorem 2

The optimal interpolation kernels $\tilde{v}_1(\boldsymbol{t}), \ldots, \tilde{v}_K(\boldsymbol{t})$ must satisfy the orthogonality principle

$$E[(x(\boldsymbol{t}) - \hat{x}(\boldsymbol{t}))y_k(\boldsymbol{m})] = 0, \quad \forall \boldsymbol{m} \in \mathbb{Z}^d, \forall k \in [1, K]. \tag{64}$$

Substituting (23), (64) becomes

$$\sum_{\ell=1}^{K} \sum_{\boldsymbol{n} \in \mathbb{Z}^d} R_{yy}^{k,\ell}(\boldsymbol{n} - \boldsymbol{m}) \tilde{v}_\ell(\boldsymbol{t} - \boldsymbol{n}) = R_{xy}^k(\boldsymbol{t} - \boldsymbol{m}), \tag{65}$$

for all $\boldsymbol{m} \in \mathbb{Z}^d$ and $k \in [1, K]$. With a change of variables $\boldsymbol{\tau} = \boldsymbol{t} - \boldsymbol{m}$ and $\boldsymbol{n}' = \boldsymbol{n} - \boldsymbol{m}$, (65) can be written as

$$\sum_{\ell=1}^{K} \sum_{\boldsymbol{n}' \in \mathbb{Z}^d} R_{yy}^{k,\ell}(\boldsymbol{n}') \tilde{v}_\ell(\boldsymbol{\tau} - \boldsymbol{n}') = R_{xy}^k(\boldsymbol{\tau}), \ \forall k \in [1, K]. \tag{66}$$

Taking the CSFT of both sides, we get

$$\sum_{\ell=1}^{K} \tilde{V}_\ell(\boldsymbol{\omega}) \sum_{\boldsymbol{n} \in \mathbb{Z}^d} \Gamma_{yy}^{k,\ell}(\boldsymbol{\omega} - 2\pi\boldsymbol{n}) = \Gamma_{xy}^k(\boldsymbol{\omega}), \ \forall k \in [1, K]. \tag{67}$$

Put in matrix form, (67) leads to (24).

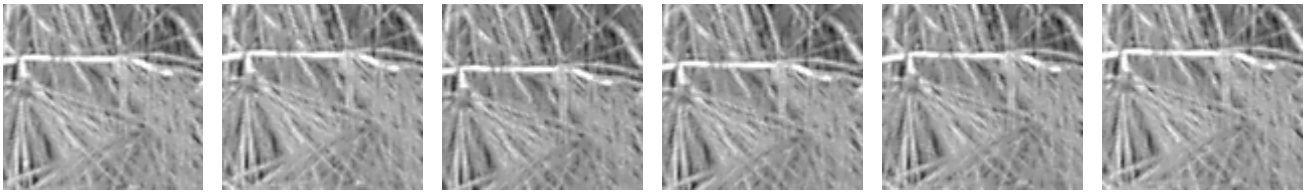
Appendix B
Proof of Theorem 3

Using the orthogonality principle, the error (59) of the LLMSE estimate $\hat{x}(\boldsymbol{t})$ is given by

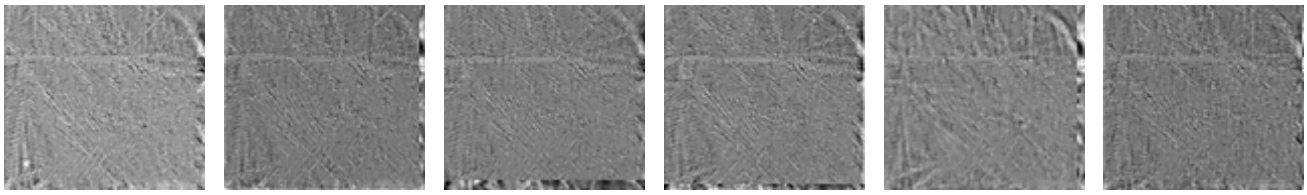
$$\begin{aligned} e(\boldsymbol{t}, \boldsymbol{\eta}) &= E[(x(\boldsymbol{t}) - \hat{x}(\boldsymbol{t}))(x(\boldsymbol{\eta}) - \hat{x}(\boldsymbol{\eta}))] \\ &= E[(x(\boldsymbol{t}) - \hat{x}(\boldsymbol{t}))x(\boldsymbol{\eta})]\,. \end{aligned} \tag{68}$$

Substituting the expression (23) for $\hat{x}(\boldsymbol{t})$, (68) becomes

$$\begin{aligned} e(\boldsymbol{t}, \boldsymbol{\eta}) &= E\left[\left(x(\boldsymbol{t}) - \sum_{k=1}^{K} \sum_{\boldsymbol{n} \in \mathbb{Z}^d} y_k(\boldsymbol{n}) \tilde{v}_k(\boldsymbol{t} - \boldsymbol{n})\right) x(\boldsymbol{\eta})\right] \\ &= R_{xx}(\boldsymbol{t} - \boldsymbol{\eta}) - \sum_{k=1}^{K} \sum_{\boldsymbol{n} \in \mathbb{Z}^d} \tilde{v}_k(\boldsymbol{t} - \boldsymbol{n}) R_{xy}^k(\boldsymbol{\eta} - \boldsymbol{n}). \end{aligned} \tag{69}$$

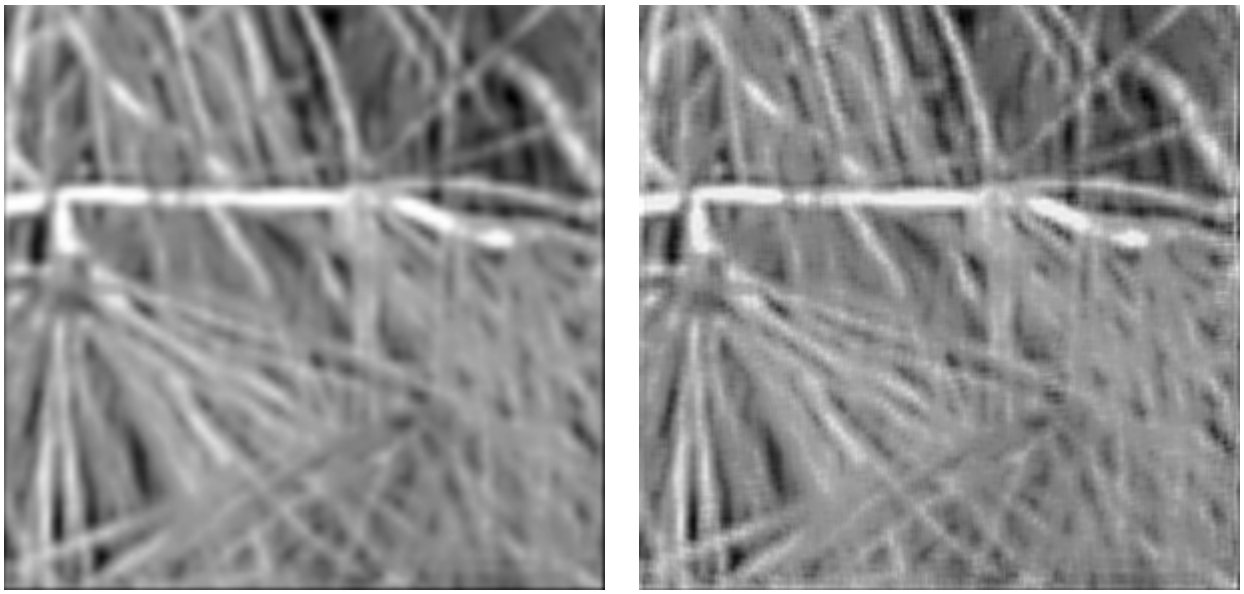


(a) Original low-resolution frames.



(b) Corrected low-resolution frames.

Fig. 7: (a) A set of low-resolution images $c_1[\boldsymbol{n}], \ldots, c_6[\boldsymbol{n}]$, $\boldsymbol{n} \in \mathbb{Z}^2$. (b) The corresponding set of corrected images $d_1[\boldsymbol{n}], \ldots, d_6[\boldsymbol{n}]$, $\boldsymbol{n} \in \mathbb{Z}^2$.



(a) (b)

Fig. 8: Resolution enhancement by a factor of $\Delta = 3$ with the SSR method ($D_x = 0$, SNR $= 10$dB). (a) Without digital correction. (b) With digital correction.

The kernels $\tilde{v}_1(\boldsymbol{t}), \ldots, \tilde{v}_K(\boldsymbol{t})$ are those minimizing the MSE, namely those given in (24).

ACKNOWLEDGMENT

The authors would like to thank Ms. Shira Avidar and Ms. Shlomit Argov for implementation and experimentation with the SSR algorithm.

REFERENCES

[1] 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.

[2] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *Int. J. Imaging Syst. Technol.*, vol. 14, no. 2, 2004.

[3] E. Shechtman, Y. Caspi, and M. Irani, "Increasing space-time resolution in video," *Lecture Notes in Computer Science*, pp. 753–768, 2002.

[4] S. Borman and R. L. Stevenson, "Super-resolution from image sequences–A review," in *1998 Midwest Symp. Circuits and Systems*, 1998, pp. 374–378.

[5] K. Sauer and J. Allebach, "Iterative reconstruction of bandlimited images from nonuniformly spaced samples," *IEEE Trans. Circuits Syst.*, vol. 34, no. 12, pp. 1497–1506, 1987.

[6] S. P. Kim, N. K. Bose, and H. M. Valenzuela, "Recursive reconstruction of high resolution image from noisy undersampled multiframes," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 6, pp. 1013–1027, 1990.

[7] S. P. Kim and W. Y. Su, "Recursive high-resolution reconstruction of blurred multiframe images," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1991, pp. 2977–2980.

[8] K. Aizawa, T. Komatsu, and T. Saito, "A scheme for acquiring very high resolution images using multiple cameras," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing.*, vol. 3, 1992.

[9] H. Ur and D. Gross, "Improved resolution from subpixel shifted pictures," *CVGIP: Graphical Models and Image Processing*, vol. 54, no. 2, pp. 181–186, 1992.

[10] N. K. Bose, H. C. Kim, and H. M. Valenzuela, "Recursive total least

squares algorithm for image reconstruction from noisy, undersampled frames," *Multidimensional Systems and Signal Processing*, vol. 4, no. 3, pp. 253–268, 1993.

[11] H. Shekarforoush and R. Chellappa, "Data-driven multichannel super-resolution with application to video sequences," *J. Opt. Soc. Am. A*, vol. 16, no. 3, pp. 481–492, 1999.

[12] A. M. Tekalp, M. K. Ozkan, and M. I. Sezan, "High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 3, 1992.

[13] S. Ramani, D. Van De Ville, T. Blu, and M. Unser, "Nonideal sampling and regularization theory," *IEEE Transactions on Signal Processing*, vol. 56, no. 3, p. 1055, 2008.

[14] Y. C. Eldar and T. Michaeli, "Beyond bandlimited sampling," *IEEE Signal Process Mag.*, no. 3, pp. 48–68, May 2009.

[15] C. A. Glasbey, "Optimal linear interpolation of images with known point spread function," *Scandinavian Image Analysis Conference–SCIA-2001*, pp. 161–168, 2001.

[16] S. Ramani, D. Van De Ville, and M. Unser, "Non-ideal sampling and adapted reconstruction using the stochastic Matérn model," *Proc. Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, 2006.

[17] T. Michaeli and Y. C. Eldar, "High rate interpolation of random signals from nonideal samples," *IEEE Transactions on Signal Processing*, vol. 57, pp. 977–992, March 2009.

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

[19] P. Cheeseman, B. Kanefsky, R. Kraft, J. Stutz, and R. Hanson, "Super-resolved surface reconstruction from multiple images," Tech. Rep., December FIA-94-12, NASA, Ames Research Center, Moffett Field, CA, 1994.

[20] R. C. Hardie, K. J. Barnard, and E. E. Armstrong, "Joint MAP registration and high-resolution image estimation using a sequence of undersampled images," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1621–1633, 1997.

[21] R. R. Schultz and R. L. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 996–1011, 1996.

[22] M. B. Matthews, "On the linear minimum-mean-squared-error estimation of an undersampled wide-sense stationary random process," *IEEE Trans. Signal Process.*, vol. 48, no. 1, pp. 272–275, 2000.

[23] V. Kucera, "Factorization of rational spectral matrices: a survey of methods," in *Proc. IEE Int. Conf. Contr., Edinburgh, U.K.*, 1991, pp. 1074–1078.

[24] M. Unser, A. Aldroubi, and M. Eden, "Fast B-spline transforms for continuous image representation andinterpolation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 3, pp. 277–285, 1991.

[25] E. E. Tyrtyshnikov and E. E. Tyrtyshnikov, *A brief introduction to numerical analysis*. Birkhauser, 1997.

[26] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 2000.

[27] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1167–1183, 2002.

[28] D. Robinson and P. Milanfar, "Statistical performance analysis of super-resolution," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1413–1428, 2006.

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

[30] S. Ramani, D. Van De Ville, T. Blu, and M. Unser, "Nonideal Sampling and Regularization Theory," *IEEE Trans. Signal Processing*, vol. 56, no. 3, pp. 1055–1070, 2008.

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

[32] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1327–1344, 2004.

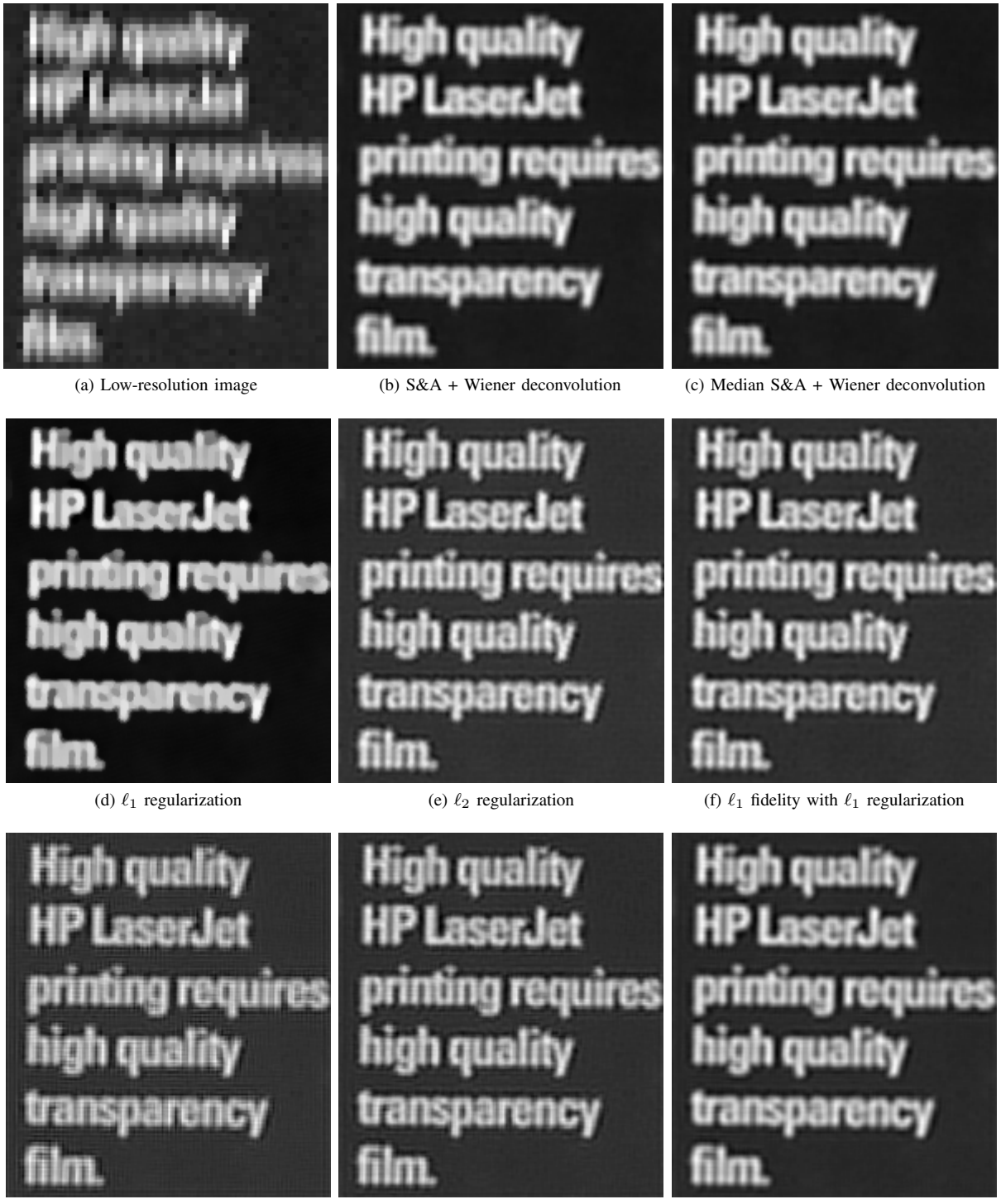
[33] "http://www.soe.ucsc.edu/~milanfar/software/sr-datasets.html."

(a) Low-resolution image (b) S&A + Wiener deconvolution (c) Median S&A + Wiener deconvolution

(d) ℓ_1 regularization (e) ℓ_2 regularization (f) ℓ_1 fidelity with ℓ_1 regularization

(g) SSR0 (h) SSR1 (i) SSR2

Fig. 9: Resolution enhancement by a factor of $\Delta = 4$ from a set of 20 images.

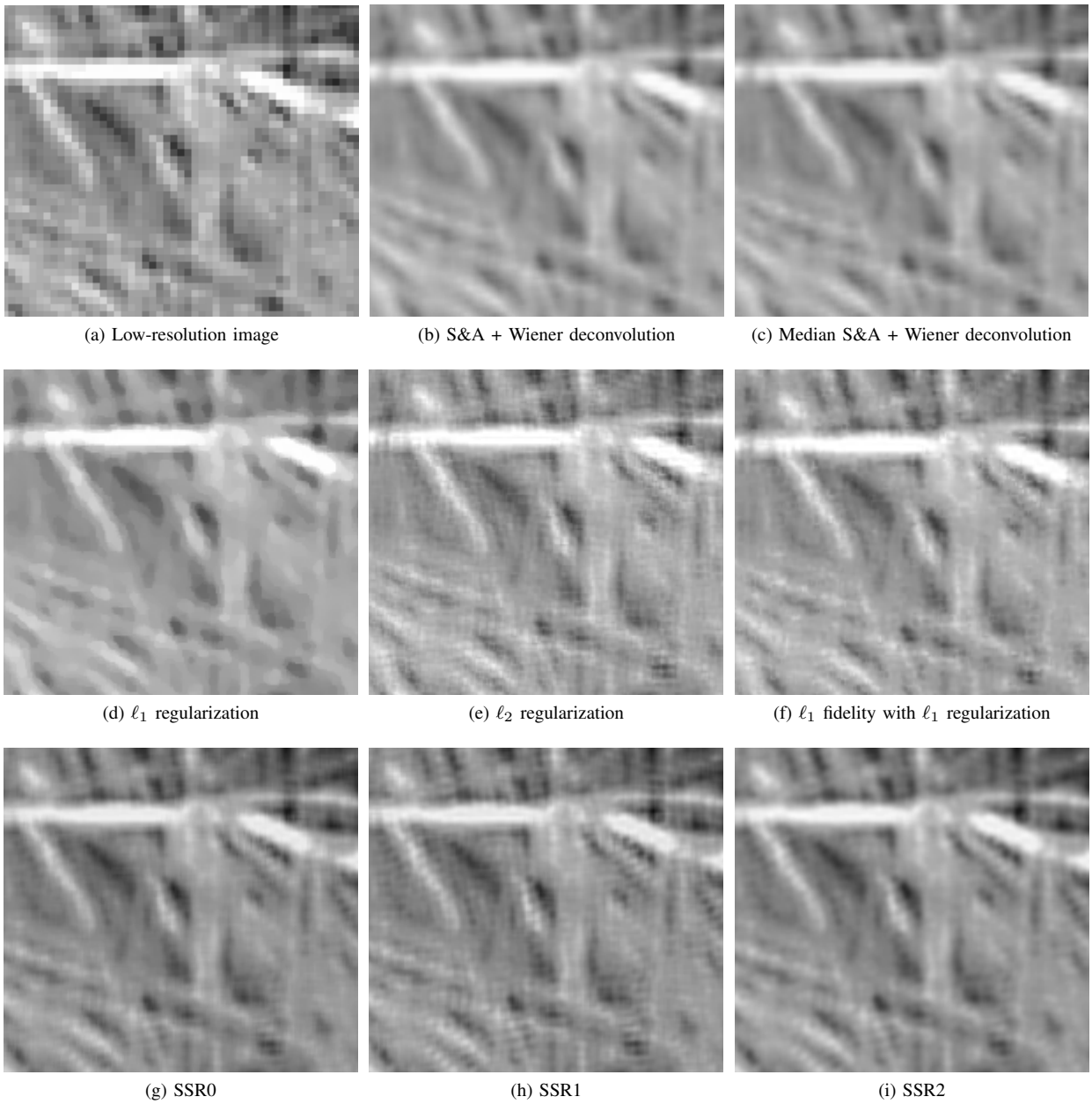


(a) Low-resolution image (b) S&A + Wiener deconvolution (c) Median S&A + Wiener deconvolution

(d) ℓ_1 regularization (e) ℓ_2 regularization (f) ℓ_1 fidelity with ℓ_1 regularization

(g) SSR0 (h) SSR1 (i) SSR2

Fig. 10: Resolution enhancement by a factor of $\Delta = 4$ from a set of 28 images.



(a) Low-resolution image (b) S&A + Wiener deconvolution (c) Median S&A + Wiener deconvolution

(d) ℓ_1 regularization (e) ℓ_2 regularization (f) ℓ_1 fidelity with ℓ_1 regularization

(g) SSR0 (h) SSR1 (i) SSR2

Fig. 11: Resolution enhancement by a factor of $\Delta = 4$ from a set of 6 images.