A dynamic texture model for imaging through turbulence

Mario Micheli*, Yifei Lou†, Stefano Soatto‡, Andrea L. Bertozzi*

micheli@math.ucla.edu, yifei.lou@ece.gatech.edu,
soatto@cs.ucla.edu, bertozzi@math.ucla.edu

*Department of Mathematics, UCLA, Los Angeles, CA 90095-1555
†School of Electrical and Computer Engineering, Georgia Inst. of Technology, Atlanta, GA 30332
‡Department of Computer Science, UCLA, Los Angeles, CA 90095-1596

Abstract

In this paper we address the problem of recovering an image from a sequence of distorted versions of it, where the distortion is caused by what is commonly referred to as ground-level turbulence. In mathematical terms, such distortion can be seen as the cumulative effect of a time-dependent anisoplanatic (space-variant) blur and a time-dependent deformation of the image domain. We introduce a statistical dynamic model for the generation of turbulence based on the dynamic texture approach. We expand the model to include the unknown image as part of the unobserved state and apply Kalman filtering to estimate such state. This operation yields a blurry image where the blurring kernel is effectively space invariant: this is verified empirically and described analytically in terms of the class of employed kernels. Applying blind nonlocal Total Variation (NL-TV) deconvolution yields a sharp final result.

1 Introduction

The phenomenon that is commonly referred to as "turbulence" in imaging is caused by the time- and space-varying refraction index of the air which is due, among other factors, to temperature, air pressure, humidity, and wind conditions between the acquired scene and the image-capturing device. The resulting image sequence is also affected by the different and changing lighting conditions within the scene, by the actual distance between the observed objects and the camera, and by other artifacts introduced by the device itself. The the problem of reconstructing an image that is altered by ground-level atmospheric turbulence still largely unsolved. The typical situation is the one illustrated in Figure 1, that shows a frame within a sequence of images taken at the Naval Station at China Lake, in California. The image shows a silhouette and a patterned board observed at a distance of 1 km; it is apparent that the point-spread function (PSF) is time- and space-varying. The images are taken at a rate of 30 frames per second: Figure 2 shows five frames within the same sequence (two consecutive images in the figure are in fact eight time samples apart). One normally assumes that there is an underlying "true" static image, to be recovered, that is corrupted by the combined effect of time- and space-dependent blur, domain deformation, and additive noise.

Related work. There is a fairly large literature in optical astronomy ([10, 36, 49] and references therein) which is concerned with the problem of reconstructing images that are mostly obtained by telescopes that operate in quasi-isoplanatic atmospheric conditions. These classical methods (see also, for example, [33, 34, 38, 42]) are however mostly inadequate under the conditions that are typical of ground-level atmospheric turbulence, i.e. with distortion, blurring and noise that are markedly not space- nor time-invariant. A type of approach that proved popular and effective in optical astronomy, specifically in situations where the field-of-view is smaller than the isoplanatic angle (thus making the blur space-invariant) is the so-called *lucky frame* (LF) method [12, 37]: one simply chooses, among the frames that make up a sequence, the one that is the sharpest. This technique was adapted to anisoplanatic conditions by Aubailly *et al.* [2] who formulated the *lucky region fusion* (LRF) approach: by assuming that different regions within the image frame are made sharper and less distorted at random times, they design an algorithm to detect such regions and fuse them into a single high quality image. Also, they propose a criterion for choosing the lucky

(a) (b)

Figure 1: (a) Original frame. (b) Detail from original frame (in black & white).

region *size*, that is then used uniformly irrespective of region location (an adaptive lucky region size should perhaps be considered). In any case both the LF and LRF approaches suffer from the limitation that it is assumed that the image sequence is extracted from short-exposure videos, thus with limited blur, and that "reasonably" lucky frame and regions do exist: however in many situations this is a low probability event, especially when the number of available frames is limited.

Shimizu *et al.* [41] propose a three-stage process that consists of (i) a deformation-free (but blurry) image is computed from the image sequence simply by temporal averaging; (ii) each frame in the sequence is registered to the image computed in the first stage by B-spline interpolation; and (iii) a multi-frame super-resolution algorithm is used to construct a detailed image from the registered sequence. Such process, although seemingly effective, is computationally very intensive especially in estimating the deformation parameters; also, the diffusion process that is implicit in temporal averaging exacerbates the blurring. Gilles *et al.* [14] propose a scheme where the image sequence is registered to the temporal mean, then the mean is re-computed using the registered sequence, and the process is repeated until convergence; the resulting image is regularized using blind total variation deconvolution [8, 20]. They perform image registration via Large Deformation Diffeomorphic Metric Mapping (LDDMM), introduced by Miller *et al.* [3, 32, 50] which is computationally very expensive; moreover the final results depend heavily on the choice of the parameters of the registration technique and no automatic algorithm for the choice of such parameters has been formulated for this type of application. Li *et al.* [26] use Principal Component Analysis (PCA) to deblur a sequence of turbulent images, simply by taking the statistically most significant vector as their estimate for the original static image; instead, Hirsch *et al.* [21] formulate a space-variant deblurring algorithm that is seemingly computationally treatable. However neither [26] nor [21] address the issue of domain deformation and thus are only suitable when the geometric distortion is reasonably small. Tahtali *et al.* [44] model the image deformation process by attaching a spring-mass-damper oscillator model to each pixel with a random perturbation, and augment the state to include the unknown natural frequency and damping ratio at each location; they treat each pixel location and coordinate independently and perform Extended Kalman Filtering (EKF) to estimate the "true" position of each pixel and produce a static image. However: the state space dimension of such a model is so large to be untreatable; to circumvent this issue they treat each pixel location and dimension independently, which is certainly not a good approximation of the physical phenomenon taking place; by their own admission, the results depends very heavily on the initial guess of the initial value of the natural frequency at each location; finally, they only test their technique on artificial data. Zhu and Milanfar [51] register the image sequence to the mean and then reconstruct a static image by minimizing a functional with a Total Variation-type regularizing term, assuming that the blurring kernel is space-invariant; in [52] they replace the second step by an algorithm that yields a static image with supposedly space-invatiant blur by combining sharp patches in the registered sequence, which is followed by a final blind deconvolution step. Last, we should mention a so far unpublished technical report by Mao and Gilles [15] where they simultaneously register the

Figure 2: Frames from the original sequence; images in each consecutive pair are 8 samples apart.

sequence frames to a static image and regularize the latter and via split-Bregman iteration, and the results are competitive; no correlation between the image frames is assumed.

Regarding related problems, Shen and Kang [23, 24, 40] try to correct the effect of *jitter* in images, i.e. the deformation due to the loss of line synchronization pulses in video manipulation devices which results in random horizontal line displacements: the nature of the distortion is however very different from the one that we try to correct for in this paper, and the mathematical models suggested by the authors cannot be applied. More recently, Tai *et al.* [27, 45, 46] have proposed solutions to the *motion blur* problem, which occurs when parts of the image region (or its entirety) are affected by space-varying blur caused by either camera motion or the motion of the represented object; however such solutions cannot be applied to the ground-level atmospheric blur as the latter does not have comparable smoothness, local nature, and general regularity.

Our approach. None of the approaches listed above, perhaps with the sole exception of [44], consider using a model for the *temporal evolution* of the image formation process. However, simple inspection of representative data suggest that, although temporally undersampled relative to the atmospheric dynamics, the images contain significant temporal correlation that should be exploited. Thus it is natural to model the image sequence as the output of a linear, time-invariant stochastic linear system [25]. Also, the spatial redundancy of each frame within the sequence suggests that the process can be represented by a state variable whose dimension is considerably smaller than the actual size, measured in number of pixels, of an image. In this paper we use a dynamic texture [9] approach to model the image formation process as a linear stochastic system with a low-dimensional underlying state process. Such model does not rely on physical principles but is statistical in nature and the system parameters can be identified efficiently; in fact they can be recalibrated to changing atmospheric conditions using a short new training sequence. Then an alternate estimation (AE) scheme based on Kalman filtering is used to estimate the underlying scene, which is supposed to be static; this step yields an image whose blurring is mostly space-invariant (isoplanatic). Applying nonlocal Total Variation (NL-TV) schemes [13, 29] to such image produces a sharp final result.

2 Problem setup

Figure 1 suggests that what is intuitively referred to as "turbulence" can be modeled, at least in first approximation, as the combined effect of (i) a time-dependent diffeomorphic deformation of the image domain and (ii) a blur with an anisoplanatic point spread function. Such approach, also used by other authors, is somewhat artificial it that (i) and (ii) can be combined into *one* linear transformation, as we shall see below. However it is effective in terms of modeling and simulation, and suggests that the recovery procedure can be split into a registration step and a deblurring step.

From now on we shall assume that there is a "true" underlying scalar luminance $\rho : \Omega \to \mathbb{R}$ that we are trying to recover, where $\Omega \subset \mathbb{R}^2$ is the rectangular image domain; on the other hand the data is time-dependent and can be represented as a function $I : \Omega \times [0, T] \to \mathbb{R}$, for some $T > 0$. We now argue that under reasonable assumptions the effects of (i) and (ii) can be simultaneously

written as a linear mapping $\rho \mapsto I$ of the type:

$$I(x,t) = \int_\Omega K(x,y,t)\,\rho(y)\,dy, \quad x \in \Omega,\, t \in [0,T]. \tag{1}$$

The deformation of the image domain (i) can be expressed as a family of diffeomorphisms $\Omega \to \Omega$, which we write as $\varphi_1(\cdot,t) \in \mathrm{Diff}(\Omega,\Omega)$, $t \in [0,T]$, so that the image ρ undergoes the mapping

$$\text{(i)} \quad \rho \mapsto \rho^\star(x,t) := (\rho \circ \varphi_1^{-1})(x,t) = \rho\big(\varphi_1^{-1}(x,t)\big), \quad x \in \Omega,\, t \in [0,T]. \tag{2}$$

For now we assume that the blurring effect (ii) has the completely general linear form:

$$\text{(ii)} \quad \rho^\star \mapsto (\mathcal{G}\rho^\star)(x,t) := \int_\Omega \int_0^T g(x,z,t,s)\,\rho^\star(z,s)\,ds\,dz, \quad x \in \Omega,\, t \in [0,T] \tag{3}$$

for some function $g : \Omega^2 \times [0,T]^2 \to \mathbb{R}$, so combining the two yields

$$I(x,t) = \int_\Omega \int_0^T g(x,z,t,s)\,\rho(\varphi_1^{-1}(z,t))\,ds\,dz = \int_\Omega \Big\{ \int_0^T g(x,\varphi_1(y,s),t,s)\ \big|J_y\varphi_1(y,s)\big|\,ds \Big\} \rho(y)\,dy,$$

where $J_y\varphi_1(y,t)$, $y \in \Omega$, $t \in [0,T]$ is the determinant of the Jacobian matrix of the vector-valued map $y \mapsto \varphi_1(y,t)$, for fixed t. So (1) holds with $K(x,y,t)$ given by the expression in braces.

We now introduce the assumption that the kernel g of the linear transformation (3) takes the still rather general form:

$$g(x,y,t,s) = k(x - \varphi_2(y,t))\,\delta(t-s), \tag{4}$$

where δ is Dirac's delta function, $k : \Omega \to \mathbb{R}$ is some isotropic kernel and $\varphi_2(\cdot,t) \in \mathrm{Diff}(\Omega,\Omega)$, $t \in [0,1]$ is another set of diffeomorphisms. In other words, we assume that the blurring is time-invariant and that the point-spread function $k(\varphi_2(\cdot,t))$, at each time $t \in [0,T]$, is a "warped" version of k — that is, it is obtained from k by deforming diffeomorphically its domain (note that the level curves of k and $k \circ \varphi_2$ are topologically equivalent). Under the assumption (4) the transformation $\rho \mapsto I$ takes the form:

$$I(x,t) = \int_\Omega k\big(x - \varphi_2(\varphi_1(y,t),t)\big)\big|J_y\varphi_1(y,t)\big|\,\rho(y)\,dy.$$

It is reasonable to assume that $\frac{1}{T}\int_0^T |J_y\varphi_1(y,t)|\,dt = 1$ for all $y \in \Omega$ (i.e. the diffeomorphisms $\varphi_1(\cdot,t)$, $t \in [0,1]$ do not, *on average*, increase or decrease the area of subsets of Ω) and it is always the case that $\int_\Omega |J_y\varphi_1(y,t)|\,dy = \mathrm{Area}(\Omega)$ for all $t \in [0,1]$. So if we assume that $|J_y\varphi_1(y,t)| \simeq 1$ (which is equivalent to assuming that φ_1 causes limited compression/expansion, i.e. that $\partial\varphi_1/\partial t$ is almost divergence-free, or incompressible; we make no assumptions on φ_2) we can also write

$$I(x,t) \simeq \int_\Omega k\big(x - \varphi_2(\varphi_1(y,t),t)\big)\rho(y)\,dy = \int_\Omega k\big(x - (\varphi_2 \circ \varphi_1)(y,t)\big)\rho(y)\,dy,$$

so that

$$I\big((\varphi_2 \circ \varphi_1)^{-1}(x,t),t\big) \simeq \int_\Omega k\big(x-y\big)\rho(y)\,dy =: (\mathcal{K}\rho)(x), \tag{5}$$

which is a linear, time-independent and isoplanatic (space-invariant) transformation. The goal of this paper is precisely to formulate an algorithm for finding $\varphi := \varphi_2 \circ \varphi_1$ (i.e. for solving the *registration* problem), so to reduce the image recovery problem to a simpler *deblurring* problem.

Summary. We can separate the image recovery problem into the inference of a time-dependent domain deformation φ and the inference of an unknown isoplanatic kernel k: in other words, under the above assumptions the problem reduces to estimating the warping of the image sequence, concatenated with a blind deconvolution procedure. Note also that from a computational point of view it is convenient to estimate the domain deformation *before* the blurring kernel as the recovery of optical flow is more accurate for image for image sequences that have some degree of blurriness (many motion field recovery algorithms [4, 43, 30] in fact include a blurring pre-processing step).

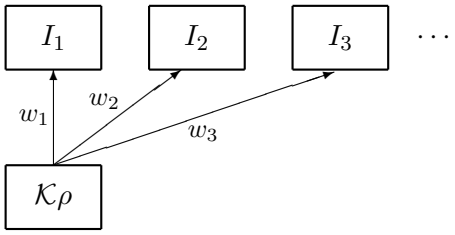


Figure 3: Relation between the original image and the data.

3 Dynamic models for ground-level turbulence

In order to introduce our algorithm we shall represent the data by the image sequence $\{I_k\}$ with $I_k(x) = I(x, k\Delta t)$, where $x \in \Omega$, $k \in \mathbb{N} := \{1, 2, 3 \ldots\}$ and Δt is the time sampling period. We assume that the data is generated by applying an unknown sequence of domain transformations $\{w_k\}$ to the function $\mathcal{K}\rho$, defined in (5); such transformations are precisely the time-sampled diffeomorphisms $w_k(x) := (\varphi_2 \circ \varphi_1)(x, k\Delta t)$, $x \in \Omega$, $k \in \mathbb{N}$, also introduced in (5). In formulas:

$$I_k(x) := (\mathcal{K}\rho)(w_k^{-1}(x,t)) \qquad \text{or simply} \qquad I_k = (\mathcal{K}\rho) \circ w_k^{-1}, \qquad x \in \Omega,\, k \in \mathbb{N}. \tag{6}$$

The data-generating model is summarized in Figure 3. Note that we can write each diffeomorphic transformation $w_k : \Omega \to \Omega$ as $w_k(x) = x + W_k(x)$, $x \in \Omega$, $k \in \mathbb{N}$, where the vector field $W_k : \Omega \to \mathbb{R}^2$ indicates the *optical flow* [22] induced by the map w_k. We say that I_k is obtained by "warping" the image $\mathcal{K}\rho$ via the flow W_k (or the corresponding transformation w_k).

In the subsections that follow we shall describe a dynamic model for the temporal evolution of the motion field, that will later be used for the simultaneous estimation of both the sequence $\{w_k\}_{k\in\mathbb{N}}$ and the blurred image $\mathcal{K}\rho$. In the existing literature on the problem the approaches that make use of a dynamic (temporal) model for turbulence are rare—the reason being that *physical* models for such phenomena are rather complex and impractical to use in a computational setting. We shall instead resort to a completely *statistical* model that relies on the dynamic texture approach introduced in [9].

3.1 Dynamic Texture

Here we review the basic dynamic texture model that will later be adapted to optical flow. Doretto *et al.* [9] introduced a statistical model to artificially synthesize time-dependent image texture for a sequence of images $\{I_k \,|\, k \in \mathbb{N}\}$ by employing a dynamic model of the type:

$$\begin{cases} \xi_{k+1} = A\xi_k + q_k & q_k \sim \mathcal{N}(0, Q) \\ I_k = C\xi_k + r_k & r_k \sim \mathcal{N}(0, R) \end{cases} \qquad k \in \mathbb{N}, \tag{7}$$

where $\xi_k \in \mathbb{R}^n$ is a low-dimensional state (not to be confused with the coordinates $x \in \Omega \subset \mathbb{R}^2$ that we used in section 2) whose temporal evolution is determined by the first linear equation while $I_k \in \mathbb{R}^m$ is the observed image (with $m = m_1 m_2$ for $m_1 \times m_2$ images); typically texture has a high degree of spatiotemporal redundancy thus the underlying state, which somehow summarizes the evolution of texture, has a much lower dimension than the actual image, so $n \ll m$. In the above equations q_k and r_k, $k \in \mathbb{N}$, are independent, additive, white Gaussian noise terms with constant covariance matrices $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$; $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{m\times n}$ are also constant.

The problem of *system identification* consists in estimating the model parameters A, C, Q, R and the initial state ξ_1 that "best fit" the measurements $\{I_1, \ldots, I_\tau\}$, with $\tau \geq n$, in a way that we shall specify shortly. First of all note that such problem does not have a unique solution since there are infinitely many matrices that give rise to the same sample path (this is easily seen by choosing

an arbitrary $M \in GL_n(\mathbb{R})$, i.e. a real invertible $n \times n$ matrix, and considering a *new* system of the type (7) where A, C, Q, R and the initial state ξ_1 are substituted by, respectively, MAM^{-1}, CM^{-1}, MQM^{-1}, RM^{-1} and $M\xi_1$; for a given realization of the Gaussian noise terms $\{q_k\}$ and $\{r_k\}$ the two dynamical systems generate the same data sequence $\{I_k\}$); we will impose a condition that implies uniqueness. Ideally, one would want to solve the following maximum likelihood problem: given measurements $\{I_1, \ldots, I_\tau\}$ (*training sequence*), with $\tau \gg n$, compute the estimates

$$\left(\widehat{A}, \widehat{C}, \widehat{Q}, \widehat{R}, \widehat{\xi}_1\right) := \arg \max_{A,C,Q,R,\xi_1} p(I_1, \ldots, I_\tau), \tag{8}$$

where p is the joint probability distribution of the first τ measurements that is induced by the model (7). However, given the high dimension of the measurements $\{I_k\}_{k=1}^{\tau}$, for computational reasons the known asymptotically optimal methods for systems identification (see, for example, [28] and [48]) cannot be used. The technique described in [9], which is briefly summarized and discussed below, is a fast and efficient algorithm that provides a sub-optimal solution to (8).

Summary of the algorithm. We will make the following assumptions on the matrix $C \in \mathbb{R}^{m \times n}$ (which is "long and thin" since $m \gg n$):

$$\operatorname{rank}(C) = n \qquad \text{and} \qquad C^T C = \mathbb{I}_n \tag{9}$$

where the superscript ^{T} indicates transposition and $\mathbb{I}_n$ is the $n \times n$ identity matrix, so we assume that the columns of C are linearly independent and orthonormal; this condition results in a unique solution for the identification criterion that we consider below.

Consider the three matrices:

$$I_1^\tau := [I_1, I_2, \ldots, I_\tau] \in \mathbb{R}^{m \times \tau}, \qquad \xi_1^\tau := [\xi_1, \xi_2, \ldots, \xi_\tau] \in \mathbb{R}^{n \times \tau}, \qquad r_1^\tau := [r_1, r_2, \ldots, r_\tau] \in \mathbb{R}^{m \times \tau}.$$

Note that by (7) we have that $I_1^\tau = C\xi_1^\tau + r_1^\tau$. Now let $I_1^\tau = U\Sigma V^T$ be the Singular Value Decomposition (SVD) [19] of the matrix I_1^τ, where $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{\tau \times \tau}$, with $U^T U = \mathbb{I}_m$ and $V^T V = \mathbb{I}_\tau$; the matrix $\Sigma \in \mathbb{R}^{m \times \tau}$ is diagonal and its diagonal elements $\{\sigma_i \,|\, i = 1, \ldots, \min(m, \tau)\}$, known as the *singular values* of I_1^τ, are real and non-negarive. We remind the reader that the Frobenius inner product of two matrices G and H (of the same size) is $\langle G, H \rangle_F := \sum_{i,j} G_{ij} H_{ij} = \operatorname{trace}(G^T H)$; it induces a norm that we shall indicate with $\|\cdot\|_F$. We now consider the best estimates of the matrix C and ξ_1^τ with respect to the Frobenius norm, $\left(\widehat{C}(\tau), \widehat{\xi_1^\tau}\right) := \arg\min_{C, \xi_1^\tau} \|r_1^\tau\|_F = \arg\min_{C, \xi_1^\tau} \|I_1^\tau - C\xi_1^\tau\|_F$; it follows from the fixed rank approximation property [19] that

$$\widehat{C}(\tau) = \text{first } n \text{ columns of } U \qquad \text{and} \qquad \widehat{\xi_1^\tau} = \text{first } n \text{ rows of } \Sigma V.$$

We write the matrix $\widehat{\xi_1^\tau} \in \mathbb{R}^{n \times \tau}$ as $\widehat{\xi_1^\tau} = [\widehat{\xi}_1, \widehat{\xi}_2, \ldots, \widehat{\xi}_\tau]$. As far as the "system matrix" $A \in \mathbb{R}^{n \times n}$ is concerned, we take $\widehat{A}(\tau) := \arg\min_A \left\|\widehat{\xi_2^\tau} - A\widehat{\xi_1^{\tau-1}}\right\|_F$, where $\widehat{\xi_2^\tau} := [\widehat{\xi}_2, \ldots, \widehat{\xi}_\tau] \in \mathbb{R}^{n \times (\tau-1)}$; this estimate can be expressed in closed form as

$$\widehat{A}(\tau) = \widehat{\xi_2^\tau}\left(\widehat{\xi_1^{\tau-1}}\right)^+,$$

where $^+$ indicates the pseudoinverse matrix. Finally, the covariance matrix Q can be estimated as

$$\widehat{Q}(\tau) := \frac{1}{\tau - 2} \sum_{k=1}^{\tau-1} \left[\widehat{\xi}_{k+1} - \widehat{A}(\tau)\widehat{\xi}_k\right]\left[\widehat{\xi}_{k+1} - \widehat{A}(\tau)\widehat{\xi}_k\right]^T.$$

The covariance matrix R can be estimated in a similar fashion; however, due to its large size $(m \times m)$, this is rarely calculated. We refer the reader to [9] for more details: we shall just observe that the condition (9) makes the estimate for C uniquely determined up to a change of sign of C and x; the paper also contains Matlab code. Note that the sequence $\widehat{I}_k := \widehat{C}(\tau)\widehat{\xi}_k$, $k = 1, \ldots, \tau$ can be interpreted as a *denoised* version of the original training sequence.

Remark. The values taken by a grayscale image are nonnegative, whence for computational reasons one normally does not implement the above procedure for the actual image sequence but rather from its *deviation for the temporal mean* $\overline{I}(x) := \frac{1}{\tau}\sum_{k=1}^{\tau} I_k(x)$, $x \in \Omega$, i.e. one writes $\delta I_k(x) := I_k(x) - \overline{I}(x)$, $x \in \Omega$ and considers δI_k as the output of the dynamical system (7).

3.2 Adaptation of dynamic texture model to optical flow

The dynamic model described above can be used to create a (once again, purely statistical) temporal model for the evolution of *optical flow* W_k, $k \in \mathbb{N}$, by treating it as the output of a similar system:

$$\begin{cases} \xi_{k+1} = A\xi_k + q_k & q_k \sim \mathcal{N}(0, Q) \\ W_k = C\xi_k + r_k & r_k \sim \mathcal{N}(0, R) \end{cases} \qquad k \in \mathbb{N}, \tag{10}$$

where $\xi_k \in \mathbb{R}^n$, $W_k \in \mathbb{R}^{2m}$ ($m = m_1 m_2$ for $m_1 \times m_2$ images, and $2m \gg n$), $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{2m\times n}$. Once again $q_k \sim \mathcal{N}(0, Q)$, with $Q \in \mathbb{R}^{n\times n}$ and $r_k \sim \mathcal{N}(0, R)$, with $R \in \mathbb{R}^{m\times m}$. The vector W_k may be written by stacking the two m-dimensional vectors W_k^1 and W_k^2, that respectively represent the first and second dimensions of the motion field. In formulas, we write

$$W_k = \begin{bmatrix} W_k^1 \\ W_k^2 \end{bmatrix} = \begin{bmatrix} C^1 \\ C^2 \end{bmatrix} \xi_k + r_k \tag{11}$$

where matrices C^1 and C^2 are both in $\mathbb{R}^{m\times n}$.

In principle, once the parameters have been identified, system (10) can be treated as a dynamic model for the evolution of the flow $\{W_1, \dots, W_\tau\}$ for a particular training sequence of *images* $\{I_1, \dots, I_\tau\}$. However a problem arises in the identification of the model parameters, in that once a training sequence of images has been fixed, the "true" sequence of *flows* $W_1, \dots W_\tau$ that generates such images is not known because the original image ρ is not known either (in fact, estimating ρ and the sequence of flows is the goal of this paper). But training sequence must be chosen in order to identify the system parameters A, C, Q, R, and ξ_1^τ. A way to proceed is to consider some estimate $\widetilde{\rho}$ of the blurred true image $\mathcal{K}\rho$ and recover the sequence of optical flows from such estimate and the real data $\{I_1, \dots, I_\tau\}$; then this estimated sequence of optical flows can be used to identify the system parameters of (10) using the Dynamic Texture techniques summarized in §3.1. For example, the *temporal mean* $\overline{I}(x) := \frac{1}{\tau}\sum_{k=1}^{\tau} I_k(x)$, $x \in \Omega$ could be used for $\widetilde{\rho}$. The temporal mean is often used as an initial estimate for the original image ρ in most of the algorithms that have been proposed to solve the turbulence problem; however, it is very blurry and consequently the estimated optical flows can be conspicuously inaccurate. A better choice for $\widetilde{\rho}$ is what we call the *centroid*, described below.

The centroid. In Euclidean geometry the *centroid* of a set of points $\{p_1, \dots, p_N\}$ whose coordinates are known is given by $c = \frac{1}{N}\sum_{i=1}^{N} p_i$ (the points p_i may be, for example, the vertices of a polygon in $\mathbb{R}^2$). If only the positions of the points *with respect to one of them*, e.g. without loss of generality p_1, are known, i.e. if only vectors $\overrightarrow{p_1 p_i}$, $i = 2, \dots N$ are known, then we can still compute the position of c *with respect to* p_1 by calculating

$$\overrightarrow{p_1 c} = \frac{1}{N}\sum_{i\neq 1} \overrightarrow{p_1 p_i}\,, \tag{12}$$

see Figure 4(b). Also, it is immediate to verify that $\sum_{i=1}^{N} \overrightarrow{cp_i} = 0$, and that $\overrightarrow{qc} = \frac{1}{N}\sum_{i=1}^{N} \overrightarrow{qp_i}$ for any point q; see Figure 4(c). These elementary considerations lead to the following idea. Suppose we want to compute the "centroid" of the set of N images $\{I_i(x), x \in \Omega\}_{1\le i\le N}$. Thinking of them as "points" and of the optical flow between two images as a "vector" between them, we can compute the optical flow between the reference image I_1 and the rest of them, i.e. the N "vectors" $\overrightarrow{I_1 I_i}(x)$, $x \in \Omega$, $i = 1, \dots, N$; so, treating optical flow as an element of a linear space, we compute

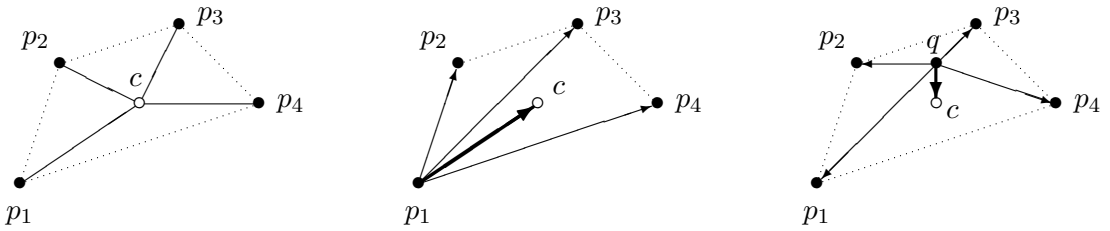


Figure 4: (a) The centroid in Euclidean space. (b) $\overrightarrow{p_1 c}$ (thick) is obtained by averaging vectors $\overrightarrow{p_1 p_i}$ (thin). (c) The "correction" $\overrightarrow{qc}$ (thick) is obtained by averaging vectors $\overrightarrow{qp_i}$ (thin).

the average flow $\overrightarrow{I_1 c}$ using equation (12) and then we warp I_1 via the flow $\overrightarrow{I_1 c}$ to finally get the "centroid" image c of the data set.

Note however that in the case of images the procedure is *not* exact since in reality the recovered optical flows do not belong to a linear space[1] so the resulting c will actually depend on the choice of the initial starting image, and in fact $\sum_{i=1}^{N} \overrightarrow{cI_i}$ will not be equal to zero. To correct this, the following iterative algorithm can be implemented: (1) a first estimate of the centroid is computed with the procedure described above, and we shall call this q; (2) the *correction* flow $\overrightarrow{qc} := \frac{1}{N} \sum_{i=1}^{N} \overrightarrow{qp_i}$ is calculated (see Figure 4(c)) and a new estimate of the centroid image c is computed by warping q via the optical flow $\overrightarrow{qc}$; (3) the previous step is repeated until convergence is achieved. Such procedure is similar to the one employed by Pennec [35] for computing the Karcher (or intrinsic) mean of a distribution on a curved manifold.

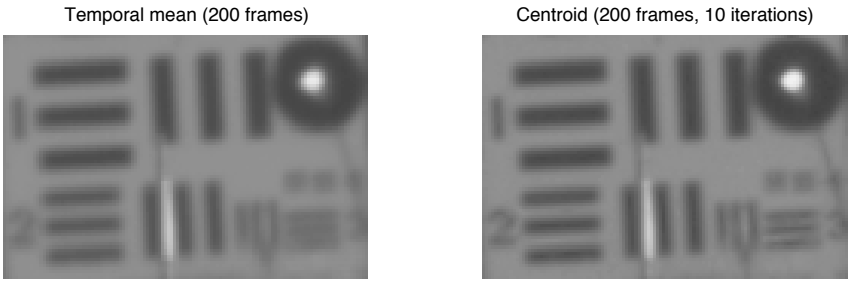
Figure 5 compares the temporal mean of a detail of the original image sequence with the centroid computed with respect to the first frame; optical flows were computed via a modern implementation of the Black-Anandan algorithm [4, 43] (available for download from Michael Black's website), whereas warping was performed by bicubic interpolation. Both the temporal mean and the centroid were computed using the first 200 frames of the sequence. In the case of the centroid, the iteration described above was repeated 10 times and yielded a result that is much better than the one obtained after the first iteration. Note that the edges are evidently straightened in both pictures when compared to those in Figure 1(b), but the centroid is much sharper than the temporal mean (note especially the number 3 on the bottom-right corner of the board and the bars next to it). We should also add that the computing the centroid yields a *much* sharper result than simply registering a single frame onto the temporal mean of the entire image sequence.

Identification and simulation. After the centroid is computed, it can be used as an initial estimate $\widetilde{\rho}$ of the image $\mathcal{K}\rho$ and the optical flows $\mathcal{W} := \{\widetilde{W}_1, \dots, \widetilde{W}_\tau\}$ between $\widetilde{\rho}$ and $\{I_1, \dots, I_\tau\}$ can be estimated; once this is done, the estimated flows $\mathcal{W}$ can be used to identify the parameters A, C, Q, R, and ξ_0 of the dynamical system (10). At this point the flow can be simulated (i.e. synthetized) by generating numerically the Gaussian random sequence $q_k \sim \mathcal{N}(0, Q)$, $k \in \mathbb{N}$ and inserting it into the first equation of dynamical system (10). The resulting sequence of flows $\widehat{W}_k := C\xi_k$, $k \in \mathbb{N}$ (whose first τ samples may be viewed as a denoised version of the training sequence of flows $\mathcal{W}$)

[1] If we interpret optical flow as a time-dependent *velocity* field $v(x,t), x \in \Omega, t \in \mathbb{R}$, and we assume that the deformation between two images $\varphi(x,t)$ is the solution of $\frac{\partial}{\partial t}\varphi(x,t) = v\big(\varphi(x,t),t\big)$, with $\varphi(x,0) = x$, then v does belong to a linear space (technically, it is an element of the Lie algebra of a group of diffeomorphisms). But in this paper what we intend as "optical flow" between two images is the *deviation from the identity* $\mathrm{id} = x$ of the deformation $\varphi(x,T)$ that deforms one into the other, i.e.:

$$\varphi(x,T) = x + V_k(x),\ x \in \Omega \qquad \text{where} \qquad V_k(x) = \int_0^T v\big(\varphi(x,t),t\big)\,dt$$

(here T is the time that it takes the velocity v to deform the first image). So V_k (plus the identity $\mathrm{id} = x$) belongs to the space of diffeomorphisms, which is nonlinear. But when V_k is small we can approximate it with a velocity field, so we can still treat it as an element of a linear space.



(a) (b)

Figure 5: Comparison between the temporal mean (a) and the centroid (b) of a set of images, both computed using the first 200 samples of the image sequence and 10 iterations of the algorithm.

Figure 6: Frames obtained by warping the centroid via a synthesized sequence of optical flows ($n = 50$, $\tau = 200$); the images in each consecutive pair above are 8 time samples apart.

can then be used to warp the centroid $\widetilde{\rho}$ so to visualize the resulting (synthetic) turbulent images:

$$I_k^{\text{synthetic}} := \widetilde{\rho} \circ \widehat{w}_k^{-1},\ k \in \mathbb{N} \qquad \text{with} \qquad \widehat{w}_k(x) := x + \widehat{W}_k(x),\ x \in \Omega;$$

the results are very realistic, and they are illustrated in Figure 6 for $n = 50$, $\tau = 200$.

Finally, Figure 7 compares the eigenvalues of the identified matrix $\widehat{A}(\tau)$ in (10) for $n = 50$ (size of the hidden state ξ_k) and different values of τ (length of the training sequence). In case (a), with $n = \tau = 50$, the training sequence is too short to effectively learn the parameters of the system. As a consequence the eigenvalues of $\widehat{A}(\tau)$ lie on the complex unit circle, which is typical of *periodic* systems: that is, the *synthesized* sequence of flows is an almost exact copy of the *training* sequence, repeated indefinitely (we are overfitting the data); still, the similarity between the two suggests that the spatial redundancy within a frame is high enough that the chosen state dimension $n = 50$ is enough to effectively describe the flow on the entire image domain. Moreover, the time evolution is well captured by a *linear* dynamic model. However, in order to obtain a more statistically significant dynamic model the length of the training sequence must be increased: cases (b), (c), (d) and (e) have $\tau = 100$, $\tau = 200$, $\tau = 500$ and $\tau = 1000$ respectively (while keeping $n = 50$) and it is apparent that the resulting identified systems are considerably slower than case (a), as the eigenvalues are closer to the point $z = 1$. Also note that the eigenvalue locations seem to *converge* as τ increases, which confirms our stationarity assumptions for the time evolution of the flow. For our particular sequence convergence is achieved around $\tau = 500$ (for $n = 50$) and the fact that the greatest majority of the eigenvalues are *away* from $z = 1$ confirms that *there is* temporal correlation within the sequence of flows. Warping the centroid by the synthesized sequence of flows resulting from case (e) yields an image sequence that is perhaps less "visually appealing" than the one resulting from case (a), in that it is slower and not as reminiscent of the training sequence, but the model identified using a longer training sequence is in fact more statistically significant and reliable for *prediction* purposes. Another set of eigenvalues is shown for $n = 200$ and five different values of τ in Figures 7(f)–(j): convergence once again occurs around $\tau = 500$.

Figure 8 shows precisely the (one-step) predictive power of our model. We used the first $\tau = 500$ samples to compute the centroid $\widetilde{\rho}$ and to identify the system parameters $\widehat{A}(\tau)$, $\widehat{C}(\tau)$, $\widehat{Q}(\tau)$, and $\widehat{\xi}_1^\tau$

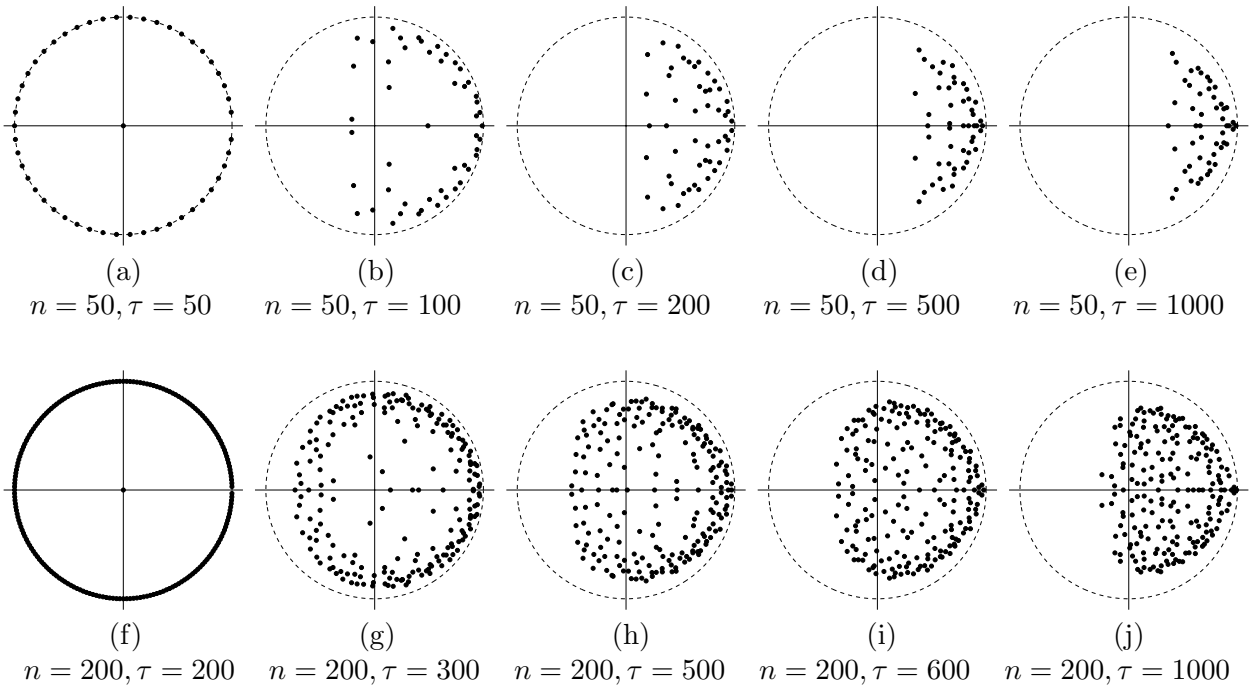


Figure 7: The eigenvalues of $\widehat{A}(\tau)$ in ten cases. The circle represents the complex curve $|z| = 1$.

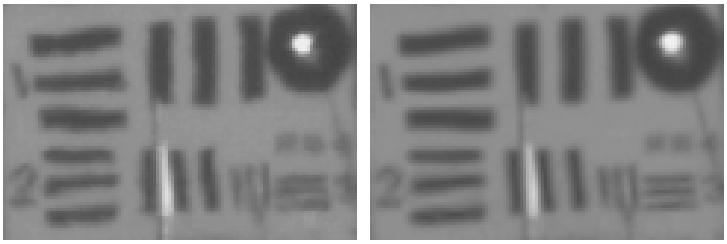


Figure 8: (a) The original frame I_{501}. (b) The one-step prediction $\widehat{I}_{501}$ (with $n = 100$).

for the system (10); parts (a) and (b) show, respectively, the actual frame I_{501} and the "predicted" image $\widehat{I}_{501}$, computed by setting $\widehat{\xi}_{501} := A\widehat{\xi}_{500}$, $\widehat{W}_{501} := C\widehat{\xi}_{501}$ and by warping the centroid $\widetilde{\rho}$ via such flow. Note that since all the eigenvalues of $\widehat{A}(\tau)$ lie inside the complex unit circle it is the case that $\widehat{A}(\tau)^k \to 0$ for $k \to \infty$, so $\widehat{\xi}_{\tau+k} := \widehat{A}(\tau)^k \widehat{\xi}_\tau \simeq 0$ and $\widehat{W}_{\tau+k} := C\widehat{\xi}_{\tau+k} \simeq 0$ for large k, therefore the estimate $\widehat{I}_{\tau+k}$ is given by the centroid itself. However it is apparent from Figure 8 that our model is effective in terms of one-step prediction, which we will use in the next section.

Remark. Instead of using a *physical* dynamical model for turbulence we have introduced a purely *statistical* one that has the further advantage of being linear, whence easily implementable for computational purposes. Furthermore one should note that the parameters can be quickly re-calibrated to changing atmospheric conditions (temperature, pressure, humidity, wind, light, et cetera) using a new training sample of images within a short time window.

4 Global dynamical model, and Kalman filtering

In the previous section we have introduced a statistical dynamic model for turbulence that can be used (in principle, in conjunction with any of the known methods) for the recovery of the underlying static image. Perhaps the most simplest scheme to employ with such dynamic model is Kalman filtering [25]. The problem that we are now addressing is in fact the recovery of the blurry version $\mathcal{K}\rho$ of the original image: so we can augment the state in (10) by adding the image $\mathcal{K}\rho$ and treating it as a hidden *constant* with respect to time, to be estimated (in the remainder of this section *we will drop the "$\mathcal{K}$" for notational convenience*). The resulting global dynamical system, that describes the time evolution of both the image and the optical flows $\{W_k\}$ and the generation of the data $\{I_k\}$ (refer to Figure 3), can be written as:

$$\rho_{k+1} = \rho_k + s_k \qquad r_k \sim \mathcal{N}(0, S) \tag{13a}$$

$$\xi_{k+1} = A\xi_k + q_k \quad q_k \sim \mathcal{N}(0, Q) \qquad\qquad k \in \mathbb{N} \tag{13b}$$

$$I_k = \rho_k \circ w_k^{-1} \tag{13c}$$

where $w_k(x) = x + W_k(x)$, $x \in \Omega \subset \mathbb{R}^2$ and, by virtue of the second of equations (10), $W_k = C\xi_k$. Once again, the state $\xi_k \in \mathbb{R}^n$ is a low-dimensional representation of the optical flow $W_k \in \mathbb{R}^{2m}$, and matrices $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{2m\times n}$ are the ones identified by the procedure described previously: they are in fact $\widehat{A}(\tau)$ and $\widehat{C}(\tau)$ from §3.2. Once again, s_k and q_k are independent, white, Gaussian additive noise terms, and it also the case that $Q = \widehat{Q}(\tau)$ from §3.2. Note that the additive noise s_k in (13a) will allow the estimate of ρ to drift away from its initial value.

We call (13a) and (13b) *state equations* and (13c) *measurement equation*; the joint state sequence $\{(\rho_k, \xi_k)\}_{k\in\mathbb{N}}$ is hidden whereas all we measure is the data sequence $\{I_k\}_{k\in\mathbb{N}}$. The tool of choice for estimating the state from the sequence of data points (images) for a dynamic model like the one above is Kalman filtering [25]. However, as we shall soon see in detail, we cannot apply directly the equations of the Kalman filter because the measurement equation is *not* linear in the *joint* state (ρ_k, ξ_k), but only in ρ_k (for fixed ξ_k). So we will have to apply an *alternate estimation* (AE) procedure at each measurement update. We define the estimates/error covariance matrix pairs:

$$\widehat{\rho}_{j|k} := \mathbb{E}\big[\rho_j | I_1 \ldots, I_k\big], \quad \Sigma_{j|k} := \mathrm{Var}\big[\rho_j | I_1 \ldots, I_k\big],$$
$$\widehat{\xi}_{j|k} := \mathbb{E}\big[\xi_j | I_1 \ldots, I_k\big], \quad P_{j|k} := \mathrm{Var}\big[\xi_j | I_1 \ldots, I_k\big].$$

where $\mathbb{E}$ and Var are the symbols for expectation and covariance, respectively. Note that W_k is also an unknown of the problem, so we can consider the estimate $\widehat{W}_{j|k} := \mathbb{E}\big[W_j | I_1 \ldots, I_k\big]$ and the corresponding error covariance matrix $\Theta_{j|k} := \mathrm{Var}\big[W_j | I_1 \ldots, I_k\big]$. However we can completely describe the estimates of W_j by the estimates of ξ_j, in fact $\widehat{W}_{j|k} = C\widehat{\xi}_{j|k}$ and $\Theta_{j|k} := CP_{j|k}C^T$.

The **Time Update (TU) equations**, or **prediction equations**, are straightforward:

$$\widehat{\rho}_{k+1|k} = \rho_{k|k} \tag{14a}$$

$$\widehat{\xi}_{k+1|k} = A\widehat{\xi}_{k+1|k} \tag{14b}$$

$$\Sigma_{k+1|k} = \Sigma_{k|k} + S \tag{14c}$$

$$P_{k+1|k} = AP_{k|k}A^T + Q \tag{14d}$$

where $S = \mathbb{E}[s_k s_k^T]$ and $Q = \mathbb{E}[q_k q_k^T]$ are the covariance matrices introduced in (13a) and (13b).

Measurement Update (MU) equations for $\boldsymbol{\rho}$. Note that the mapping $\rho \mapsto \rho\circ w^{-1}$ (for fixed w) is *linear* in ρ, so we could directly apply the equations of the Kalman filter. However since such mapping is represented by a very large matrix ($m\times m$, with $m = m_1 m_2$), for computational reasons it is convenient to rewrite the measurement equation (13c) as:

$$I_k \circ w_k = \rho_k + \zeta_k \tag{15}$$

where ζ_k is Gaussian, white, zero-mean additive noise with $\mathbb{E}[\zeta_k \zeta_k^T] = Z > 0$. Assuming that we know $w_{k+1|k+1}$ (more on this below) the measurement update equations for ρ are simply:

$$\begin{aligned} \widehat{\rho}_{k+1|k+1} &= \widehat{\rho}_{k+1|k} + L_{k+1}(I_{k+1} \circ \widehat{w}_{k+1|k+1} - \widehat{\rho}_{k+1|k}), \\ \Sigma_{k+1|k+1} &= (I - L_{k+1})\Sigma_{k+1|k}, \\ \text{where:} \qquad L_{k+1} &= \Sigma_{k+1|k}(\Sigma_{k+1|k} + Z)^{-1} \end{aligned} \tag{16}$$

is the Kalman gain. Note that in equation (16) we need the estimate $\widehat{w}_{k+1|k+1}$, i.e. $\widehat{W}_{k+1|k+1} := \mathbb{E}\big[W_{k+1}|I_1 \ldots, I_{k+1}\big] = C\widehat{\xi}_{k+1|k+1}$, but as we anticipated above we do not have $\widehat{\xi}_{k+1|k+1}$ at our disposal yet. So we may use $\widehat{\xi}_{k+1|k}$ from the time update equation (14b); in the next step (measurement update for x) we will use the value of $\widehat{\rho}_{k+1|k+1}$ from (16) to compute $\widehat{\xi}_{k+1|k+1}$, and we shall then iterate the process.

Measurement Update (MU) equations for $\boldsymbol{\xi}$. We now assume that $\rho_{k+1|k+1}$, computed with the procedure described above, is known. Whence the most natural way of performing the measurement update for x is to consider the following linear measurement equation instead of (13c):

$$W_k^{\text{meas}} = C\xi_k + r_k, \qquad r_k \sim \mathcal{N}(0, R), \tag{17}$$

which is precisely the second of equations (10), where C and R are the the matrices computed in the identification step described in §3.2, while W_k^{meas} is the measured optical flow between $\widehat{\rho}_{k|k}$ (assumed known) and I_k (data): that is, $I_k = \widehat{\rho}_{k|k} \circ (w_k^{\text{meas}})^{-1}$, with $w_k^{\text{meas}}(x) = x + W_k^{\text{meas}}(x)$, $x \in \Omega$. Such flow can be computed by using the estimation algorithm described in [4, 43]. So the measurement update equations for x are simply:

$$\begin{aligned} \widehat{\xi}_{k+1|k+1} &= \widehat{\xi}_{k+1|k} + M_{k+1}(W_{k+1}^{\text{meas}} - C\widehat{\xi}_{k+1|k}) \\ P_{k+1|k+1} &= (I - M_{k+1}C)P_{k+1|k} \end{aligned}$$

where the Kalman gain is given by

$$M_{k+1} = P_{k+1|k}C^T(CP_{k+1|k}C^T + R)^{-1}, \tag{18}$$

$$\text{or:} \qquad M_{k+1} = (C^T R^{-1} C + P_{k+1|k}^{-1})^{-1} C^T R^{-1} \tag{19}$$

where we have applied the Matrix Inversion Lemma[2] (under the assumption that $P_{k+1|k} > 0$). Note that (19) is more convenient to use than (18) because (19) involves the inversion of two $n \times n$ matrix, while (18) involves the inversion of an $m \times m$ matrix, with $m = m_1 m_2 \gg n$. The inversion of the $m \times m$ matrix R is performed, offline, only once; however, for computational purposes, it may be convenient to use a diagonal matrix R with constant diagonal elements.

Remark (**Measurement Update via an Alternating Estimation (AE) scheme**)**.** As noted before, in order to compute $\widehat{\xi}_{k+1|k+1}$ one in principle needs $\widehat{\rho}_{k+1|k+1}$, and vice versa. A reasonable approximation is to *alternate* between the two measurement updates for ρ and x: so, <u>for a fixed k</u>:

1. Perform the TU (time update) by equations (14a)–(14d).
2. Use $\xi_{k+1|k}$ instead of $\xi_{k+1|k+1}$ in the MU equations for ρ to compute $\rho_{k+1|k+1}$ and $\Sigma_{k+1|k+1}$.
3. Use $\rho_{k+1|k+1}$ in the MU equations for ξ to compute $\xi_{k+1|k+1}$ and $P_{k+1|k+1}$.
4. Use $\xi_{k+1|k+1}$ in the MU equations for ρ to compute $\rho_{k+1|k+1}$ and $\Sigma_{k+1|k+1}$.
5. Go to step **3** and iterate.

[2]Given two invertible matrices $\Sigma \in \mathbb{R}^{N\times N}$, $\Lambda \in \mathbb{R}^{M\times M}$ and a matrix $H \in \mathbb{R}^{M\times N}$, the following identity holds: $(H\Sigma H^T + \Lambda)^{-1} = \Lambda^{-1} - \Lambda^{-1}H(H^T\Lambda^{-1}H + \Sigma^{-1})^{-1}H^T\Lambda^{-1}$. This is the Matrix Inversion Lemma [25]; it is proven by verifying that multiplication of the right-hand side by $H\Sigma H^T + \Lambda$ yields the identity matrix.

Note that the iteration should be repeated a fixed number of times and *not* until convergence (because otherwise $\widehat{\rho}_{k+1|k+1}$ will converge to the current measurement I_{k+1} and $\widehat{\xi}_{k+1|k+1}$ to zero). In our implementation steps **3** and **4** are repeated twice at each Kalman iteration. Also, the Kalman gain and $\Sigma_{j|k}$, S and Z (which are all $m \times m$ matrices, with $m = m_1 m_2 \gg n$) are treated as scalars for obvious computational reasons. To achieve convergence for $\widehat{\rho}_{k|k}$, the scalar S is assigned a small value while Z is assigned a large value.

As far as other practical details of the implementation are concerned, if we have a sequence of N images $\{I_k\}_{k=1}^N$ at our disposal, we may use the first τ samples $\{I_k\}_{k=1}^\tau$ as the *training sequence* for computing the centroid and estimating the the parameters of the dynamical system (10). The remaining data $\{I_k\}_{k=\tau+1}^N$ can instead be used to run the Kalman filter-based scheme described above, using the centroid as initial estimate for ρ. By its own formulation, Kalman filtering at the kth step minimizes a mean square error, so the scheme yields a *blurry* image $\widehat{\rho}_{N|N}$, i.e. an estimate for the function $\mathcal{K}\rho$ introduced at the end of §2, since no regularization is imposed by the Kalman filtering. As we said at the end of §2 it is actually relevant at this stage to work with images that are still blurry, since this helps the estimation of optical flow needed for the measurement update of ξ. The blur in the resulting estimate $\widehat{\rho}_{N|N}$ turns out to be relatively space-invariant (isoplanatic), and a deconvolution algorithm can be applied; this final step is precisely described in the next section.

5 Deblurring via nonlocal Total Variation deconvolution

We are now left with the problem of deblurring an image with space-invariant point spread function; in this section we briefly describe the problem and summarize our method of choice. Given an unknown image $u : \Omega \to \mathbb{R}$ and an convolution kernel $k : \Omega \to \mathbb{R}$ (also referred to as point spread function, or PSF) the well-known *deconvolution problem* is the one of recovering u from the data:

$$f := k * u + n \tag{20}$$

where $*$ is the symbol for convolution and n is additive noise. The problem is ill-posed since the convolution operation typically causes the irremediable loss of some frequencies of u. So in order to make the solution to the problem unique one needs to restrict the search of u to some smaller function space. One way to proceed is to define the solution $\widehat{u}$ of the deconvolution problem as:

$$\widehat{u} = \arg\min_u \Big\{ \int_\Omega (f - k * u)^2 dx + \mu \int_\Omega U(u, \nabla u)\, dx \Big\}; \tag{21}$$

for some function U of u and its gradient. The first term is the *data-dependent term* while the second is the *regularization term*, and the constant $\mu > 0$ is the *regularization parameter*.

Two common choices for the function U are the following. Setting $U(u, \nabla u) = u^2$, so that the second integral in (21) is simply the L^2 norm of u, leads to what is known as *Tikhonov* regularization [47]; the corresponding Euler-Lagrange equation is: $k^* * (k * u - f) + \mu u = 0$, where $k^*(x) := k(-x)$, $x \in \Omega$ is the *adjoint* of the kernel. Replacing the L^2 norm with the H^1 semi-norm, i.e. choosing $U(u, \nabla u) = \|\nabla u\|^2$ defines instead the *Wiener* regularization [1] whose Euler-Lagrange equation is the PDE: $k^* * (k * u - f) - \mu \Delta u = 0$ (where Δu is the Laplacian of u). Both these methods thus lead to linear equations and one can use the Fast Fourier Transform to solve them in one step in the frequency domain. However, in order to compensate for the noise a large regularization parameter μ must be used, which causes the edges of $\widehat{u}$ to be smeared out.

To circumvent this problem Rudin, Osher and Fatemi [39] proposed the *Total Variation* (TV) functional of u as regularizing term. Applying variational techniques to

$$F_{\lambda,\mu}[u] := \frac{\lambda}{2} \int_\Omega (f - k * u)^2 dx + \mu \int_\Omega \|\nabla u\|\, dx$$

one finds that its local minimizers may be computed by gradient descent with the nonlinear PDE

$$\frac{\partial u}{\partial t} = \lambda\, k^* * (f - k * u) + \mu\, \mathrm{div}\left(\frac{\nabla u}{\|\nabla u\|}\right), \tag{22}$$

with Neumann boundary condition $\frac{\partial u}{\partial \mathbf{n}} = 0$ on $\partial\Omega$. Methods for implementing (22) are described, for example, in [7, 17, 18]; this produces denoised images with sharp edges and low-gradient regions.

An evolution of the above scheme is the so-called *nonlocal* version of Total Variation deconvolution (NL-TV) where one restores pixel patches by using information from others that are similar; see [13] and [29]. This is particularly effective when image patterns (edges, lines, corners, etc.) are repeated several times within the same image, as it is the case on the board in part (b) of Figure 1. Given the data $f(x)$, $x \in \Omega$ in model (20), first one considers a nonnegative, symmetric *weight* function $w_f(x, y)$, $(x, y) \in \Omega^2$, with the property that $w(x, y) \simeq 1$ when the patches around locations x and y are similar and $w(x, y) \simeq 0$ otherwise. For example one can choose (as in [5, 6]):

$$w_f(x, y) := \exp\Big\{ - \frac{\big(G_a * |f(x + \cdot) - f(y + \cdot)|\big)(0)}{h^2}\Big\}, \tag{23}$$

where G_a is a Gaussian kernel with standard deviation a and h is a filtering parameter; normally h is set to be the standard deviation of the noise in (20). The *nonlocal* derivative (for a fixed $y \in \Omega$) and gradient of a function $u \in L^2(\Omega, \mathbb{R})$ are defined, respectively, as:

$$\begin{aligned} \partial_y u : \Omega \to \mathbb{R}, \qquad & \text{with} \qquad (\partial_y u)(x) := [u(y) - u(x)]\sqrt{w(x, y)}, \\ \nabla_w u : \Omega \times \Omega \to \mathbb{R}, \qquad & \text{with} \qquad (\nabla_w u)(x, y) := \partial_y u(x). \end{aligned}$$

Also, given a function $v \in L^2(\Omega^2, \mathbb{R})$ we can define its *nonlocal* divergence $\mathrm{div}_w v \in L^2(\Omega, \mathbb{R})$ as the dual of the nonlocal gradient with respect to the L^2 inner product, that is, by definition: $\langle \mathrm{div}_w v, u\rangle_{L^2(\Omega,\mathbb{R})} = -\langle v, \nabla_w u\rangle_{L^2(\Omega^2,\mathbb{R})}$; it is the case [13] that this definition leads to the expression

$$(\mathrm{div}_w v)(x) = \int_\Omega \big[v(x, y) - v(y, x)\big]\sqrt{w(x, y)}\, dy, \qquad x \in \Omega.$$

Finally, the dot product of two functions $v_1, v_2 \in L^2(\Omega^2, \mathbb{R})$ is $(v_1 \cdot v_2)(x) := \int_\Omega v_1(x, y)\, v_2(x, y)\, dy$, $x \in \Omega$, and the square magnitude of v_1 is simply $|v_1|^2(x) := (v_1 \cdot v_1)(x)$, $x \in \Omega$.

With the above calculus, the NL-TV functional of a function $u \in L^2(\Omega, \mathbb{R})$ is defined as

$$J[u] := \int_\Omega |\nabla_w u|(x)\, dx = \int_\Omega \sqrt{\int_\Omega [u(x) - u(y)]^2 w(x, y)\, dy}\, dx.$$

One can show [13] that a minimizer of the functional $E_{\lambda,\mu}[u] := \frac{\lambda}{2}\int_\Omega (f - k * u)^2\, dx + \mu\, J[u]$, for fixed values of parameters $\lambda, \mu \in \mathbb{R}$, is a steady state solution of the partial differential equation

$$\begin{aligned} & u_t = \lambda\, k^* * (f - k * u) - \mu\, L[u], \qquad (24) \\ \text{where} \quad & L[u] := -\,\mathrm{div}_w\left(\frac{\nabla_w u}{|\nabla_w u|}\right) = -\int_\Omega \big[u(y) - u(x)\big]\left(\frac{1}{|\nabla_w u|(x)} + \frac{1}{|\nabla_w u|(y)}\right) w(x, y)\, dy, \end{aligned}$$

which is obtained by computing the first variation of $J[u]$. Note that the PDE (24) is formally similar to (22), except that the divergence term is substituted by its nonlocal version. We used precisely NL-TV deconvolution tosharpen the blurry image $\widehat{\rho}_{N|N}$ provided by the Kalman filter. More details, as well as our final results, are discussed in the next section.

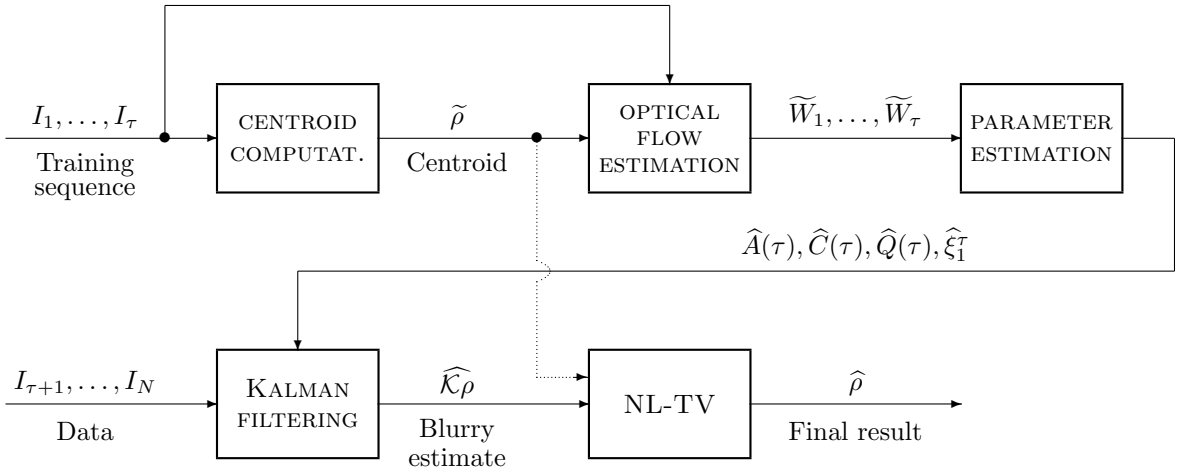


Figure 9: Flow chart of the algorithm. Note that $\widetilde{\rho}$ is also used to compute the weight for NL-TV.

6 Numerical results

Our image recovery algorithm is summarized in Figure 9. First of all, a training sequence $I_1, \ldots, I_\tau$ is used for computing the centroid, which can be used as a first estimate of the underlying image. The centroid and the training sequence itself are employed to compute a first estimate of the sequence of motion fields that generate the data, i.e. the optical flows $\widetilde{W}_1, \ldots, \widetilde{W}_\tau$ between the centroid $\widetilde{\rho}$ and the data $I_1, \ldots, I_\tau$. Following this, such sequence of motion fields is used to identify the parameters of the dynamical system (10), which models the time evolution of turbulence, by following the adapted dynamic texture procedure of §3.2. A Kalman filter is then applied to the "global" dynamical system (13), with the parameters identified at the previous step and using the centroid $\widetilde{\rho}$ and $\widehat{\xi}_\tau$ as initial estimates for ρ and x. For the reasons discussed in section 4 an alternate estimation (AE) scheme must be employed at each measurement update step. This produces a blurry image where the kernel is approximately space-invariant. Finally, a nonlocal Total Variation (NL-TV) deconvolution scheme is employed to perform the deblurring; as indicated by the dashed line connecting $\widetilde{\rho}$ to the NL-TV block, the centroid is used to compute the weight function (23). We should note that the NL-TV implementation that we are using is, for the time being, non-blind: we assume that the blurring Kernel is Gaussian, with a manually-picked standard deviation.

Figure 10 shows the result of the our algorithm with the intermediate steps (i.e. the centroid and the blurry image produced by the Kalman filtering procedure) when applied to a portion of the image sequence provided to us by the U.S. Naval Air Warfare Center at China Lake, California. We used a training sequence of only $\tau = 100$ images $\{I_1, \ldots, I_\tau\}$, fixed the dimension of the dynamical system (10) at $n = 40$ and used a data sequence of 400 images to perform the Kalman filtering. The NL-TV deconvolution algorithm produced a very crisp picture, the standard deviation of the Gaussian blurring kernel was hand-picked at 1.45, as it yielded the crispest result. We repeated the experiment with $\tau = 500$, $n = 100$ and with 400 Kalman filter steps; the results were essentially indistinguishable from the previous experiment, although calculating the centroid using 500 samples required considerably more computing time. We should also report that the final result is better than then one obtained by simply applying NL-TV directly to the centroid or the temporal mean, thus confirming that the Kalman filter algorithm indeed yields an image whose blurring is essentially isoplanatic.

Figure 11 illustrates the result of applying our scheme to an image sequence taken in the same location but at a different time of the day, with heat, pressure, wind, and lighting conditions that cause turbulence conditions so extreme that the scene is hardly visible. Parts of it are in

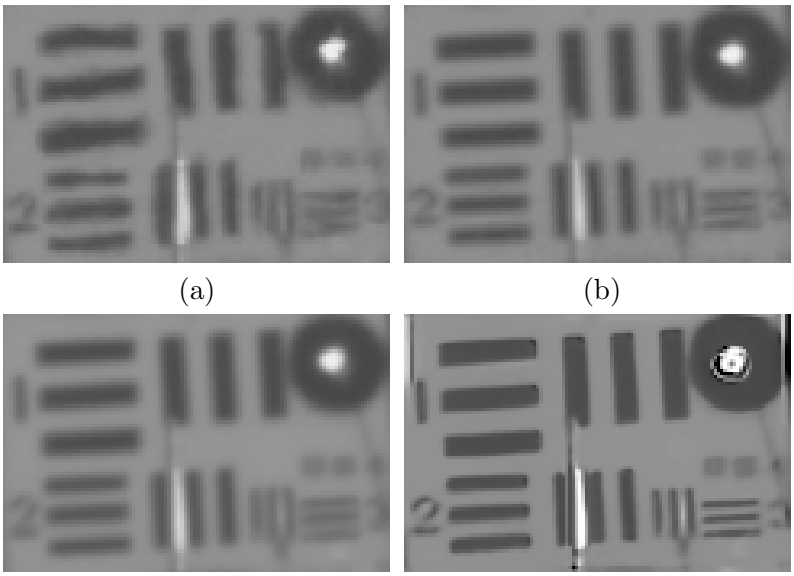


Figure 10: Comparison between: (a) a frame in the original sequence; (b) the centroid of the first 100 images; (c) the blurry image produced by 400 iterations of Kalman filtering (with $n = 40$); and (d) the image resulting from NL-TV deconvolution.

fact completely saturated by light. As expected the results are much worse than in our previous experiments, but some parts of the board (such as the circle and the thicker bars) are clearly visible after applying our algorithm. In this case we used a training sequence of $\tau = 500$ samples to compute the centroid and identify the system parameters; the state dimension of the state ξ_k in (10) was chosen to be $n = 100$, and the Kalman filter was run for 100 iterations. As far as the NL-TV scheme is concerned, we assumed that the blurring was Gaussian with a standard deviation of 3.2. Once again, we used the centroid to build the weight for the NL-TV algorithm.

7 Conclusions and future work

In this paper we have proposed a statistical method for modeling the time evolution of turbulence caused by atmospheric conditions (i.e. the combined effects of humidity, temperature, wind and lighting conditions, combined with other artifacts introduced by the image capturing device itself); we believe this is an innovation that was not yet attempted in the existing literature. We did so by fitting the parameters of a low-dimensional linear dynamical system, by adapting a dynamic texture algorithm to the evolution of optical flow. We have then used such model to formulate a method for solving the problem of recovering an image affected by ground-level turbulence,. by using a Kalman filter scheme (with alternate estimation sub-steps for the measurement update equations) that produces an image with quasi-isoplanatic blur. Finally, a nonlocal Total variation deconvolution scheme yields a sharp result.

We should point out that there are situations when the proposed method does **not** work: namely, those where a linear system of the type (10) cannot model effectively the turbulence phenomenon. This may happen for any of the following reasons, possibly combined: turbulence is not stationary in time and does not follow a coherent pattern; the optical flow does not have the spatial redundancy prerequisites for being described by a low-dimensional state; last, but certainly not least, the training sequence (i.e. the available data) is too short to correctly identify the parameters, especially when a larger state dimension is needed.

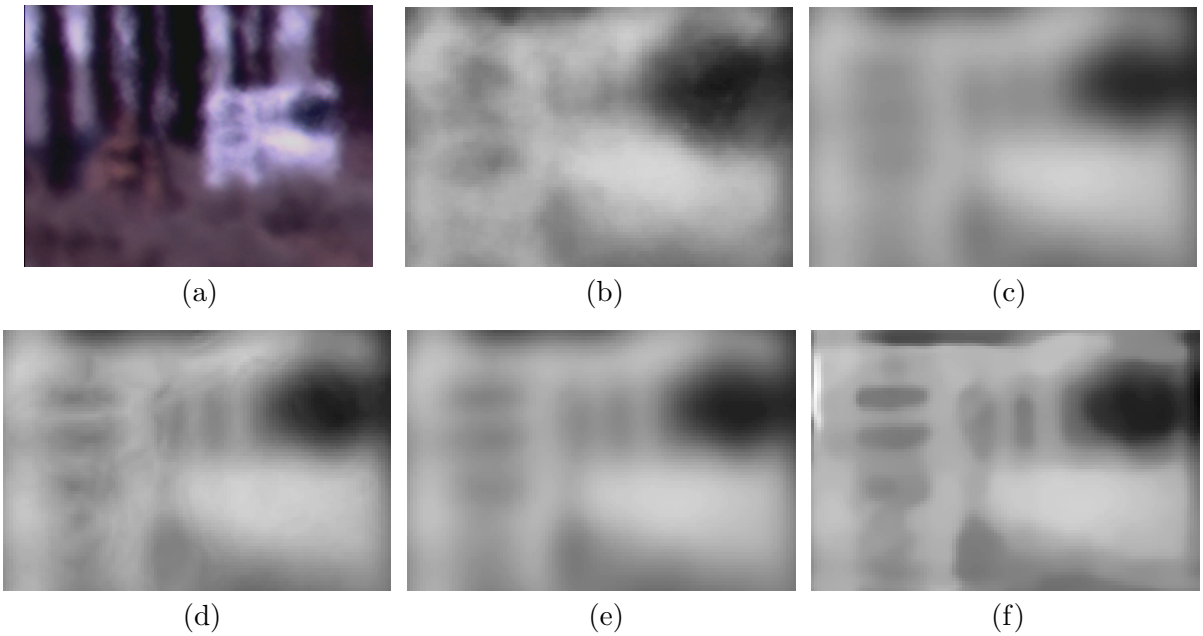


Figure 11: These figures refer to a much extreme level of turbulence: (a) a complete frame in the original sequence; (b) detail; (c) temporal mean of the first 500 frames; (d) centroid of the first 500 frames; (e) Kalman Filter output; (f) NL-TV output.

However when the conditions are such that the linear system *does* describe the data well, we have a computationally tractable model that can be used for image recovery. The model's parameters can be fitted efficiently and in fact can be quickly re-calibrated to changing atmospheric conditions; the estimate for the blurry version of the original image is computed by combining the prediction provided by the model and the actual data, using appropriate weights. In fact, while the temporal model that we use is especially suited for Kalman-filter based image estimation, one could explore the possibility of using it conjunction with other recovery methods such as those that we described in the introduction. Future work directions include but are not limited to incorporating the deblurring step into the Kalman filtering scheme and exploring the advantages of choosing different deblurring kernels, such as those known from the optics literature (see [11], [16], [31] and references therein).

8 Acknowledgements

Mario Micheli's research was partially supported by ONR grant N000140910256. Yifei Lou and Andrea Bertozzi were partially supported by ONR grants N00014101022, N000141210040 and by NSF grants DMS-0914856, DMS-1118971. Stefano Soatto was partially supported by ONR grant ONR N0001411100863. The authors would like to thank Dr. Alan Van Nevel at the U.S. Naval Air Warfare Center, Weapons Division (China Lake, California) for providing the image data. We are also deeply grateful to Stanley Osher and Jérôme Gilles of UCLA, and to Angelo Cenedese of Università di Padova (Italy) for the insightful discussions on the topic. We would also like to thank Xiaoqun Zhang, formerly at UCLA and now at Shanghai Jiaotong Univeristy, for providing nonlocal Total Variation (NL-TV) deconvolution Matlab code.

References

[1] H. C. Andrews and B. R. Hunt. *Digital Image Restoration.* Prentice Hall, Englewood Cliffs, 1977.

[2] M. Aubailly, M. A. Vorontsov, G. W. Carhart, and M. T. Valley. Automated video enhancement from a stream of atmospherically-distorted images: the lucky-region fusion approach. In *SPIE Atmospheric Optics: Models, Measurements, and Target-in-the-Loop Propagation*, San Diego, CA, Aug. 2009.

[3] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal on Computer Vision*, 61(2):139–157, 2005.

[4] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, Jan. 1996.

[5] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '05)*, San Diego, CA, 2005.

[6] A. Buades, B. Coll, and J.-M. Morel. A review of image denoising methods, with a new one. *Multiscale Modeling and Simulation*, 4(3):490–530, 2005.

[7] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20:89–97, Jan 2004.

[8] T. F. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE Transactions on Image Processing*, 7(3):370–375, Mar. 1998.

[9] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.

[10] D. L. Fried. Statistics of a geometric representation of wavefront distortion. *Journal of the Optical Society of America*, 55(11):1427–1435, Nov. 1965.

[11] D. L. Fried. Optical resolution through a randomly inhomogeneous medium for very long and very short exposures. *Journal of the Optical Society of America*, 56(10):1372–1379, Nov. 1966.

[12] D. L. Fried. Probability of getting a lucky short-exposure image through turbulence. *Journal of the Optical Society of America*, 68:1651–1657, Dec. 1978.

[13] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling and Simulation*, 7(3):1005–1028, 2008.

[14] J. Gilles, T. Dagobert, and C. De Franchis. Atmospheric turbulence restoration by diffeomorphic image registration and blind deconvolution. In *Advanced Concepts for Intelligent Vision Systems*, volume 5259 of *Lecture Notes in Computer Science*, pages 400–409. Springer-Verlag, 2008.

[15] J. Gilles and Y. Mao. Non rigid geometric distortions correction – application to atmospheric turbulence stabilization. CAM Technical Report 10-86, UCLA Department of Mathematics, Dec. 2010.

[16] J. Gilles and S. Osher. Fried deconvolution. CAM Technical Report 11-62, UCLA Department of Mathematics, Dec. 2011.

[17] D. Goldfarb and W. Yin. Second-order cone programming methods for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 27(2):622–645, 2005.

[18] T. Goldstein and S. Osher. The Split Bregman method for L^1 regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

[19] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, 3rd edition, 1996.

[20] L. He, A. Marquina, and S. Osher. Blind deconvolution using TV regularization and Bregman iteration. *International Journal of Imaging Systems and Technology*, 15(1):74–83, July 2005.

[21] M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling. Efficient filter flow for space-variant multiframe blind deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pages 607–614, San Francisco, CA, June 2010.

[22] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1–3):185–203, Aug. 1981.

[23] S. H. Kang and J. Shen. Video dejittering by bake and shake. *Image and Vision Computing*, 24(2):143–152, Feb 2006.

[24] S. H. Kang and J. Shen. Image dejittering based on slicing moments. In X.-C. Tai, K.-A. Lie, T. F. Chan, and S. Osher, editors, *Image Processing Based on Partial Differential Equations*, pages 35–55. Springer, 2007.

[25] P. R. Kumar and P. Varaiya. *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Prentice Hall, Englewood Cliffs, New Jersey, 1986.

[26] D. Li, R. M. Mersereau, and S. Simske. Atmospheric Turbulence-Degraded image restoration using principal components analysis. *IEEE Geoscience and Remote Sensing Letters*, 4(3):340–344, July 2007.

[27] H. T. Lin, Y.-W. Tai, and M. S. Brown. Motion regularization for matting motion blurred objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. To appear.

[28] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Englewood Cliffs, NJ, 2nd edition, 1999.

[29] Y. Lou, X. Zhang, S. Osher, and A. Bertozzi. Image recovery via nonlocal operators. *Journal of Scientific Computing*, 2(42), Feb. 2010.

[30] B. D. Lucas and T. Kanade. An iterative registration technique with an application to stereo vision. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, volume II, pages 674–679, Vancouver, B. C., Canada, Aug. 1981.

[31] S. Metari and F. Deschênes. A new convolution kernel for atmospheric point spread fuction applied to computer vision. In *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil, Oct. 2007.

[32] M. I. Miller and L. Younes. Group actions, homeomorphisms, and matching: a general framework. *International Journal of Computer Vision*, 41(1/2):61–84, 2001.

[33] R. G. Paxman, J. H. Seldin, M. G. Löfdahl, G. B. Scharmer, and C. U. Keller. Evaluation of phase-diversity techniques for solar-image restoration. *Astrophysical Journal*, 466:1087–1099, Aug. 1996.

[34] R. G. Paxman, B. J. Thelen, and J. H. Seldin. Phase-diversity correction of turbulence-induced space-variant blur. *Optics Letters*, 19(16):1231–1233, Aug. 1994.

[35] X. Pennec. *Probabilities and Statistics on Riemannian Manifolds: a Geometric Approach*. Research Report 5093, INRIA, January 2004.

[36] F. Roddier. *Adaptive Optics in Astronomy*. Cambridge University Press, 2004.

[37] M. C. Roggemann, C. A. Stoudt, and B. M. Welsh. Image-spectrum signal-to-noise-ratio improvements by statistical frame selection for adaptive-optics imaging through atmospheric turbulence. *Optical Engineering*, 33(10):3254–3264, 1994.

[38] M. C. Roggemann and B. M. Welsh. *Imaging through Turbulence*. CRC Press, 1996.

[39] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.

[40] J. Shen. Bayesian video dejittering by the BV image model. *SIAM Journal on Applied Mathematics*, 64(5):1691–1708, 2008.

[41] M. Shimizu, S. Yoshimura, M. Tanaka, and M. Okutomi. Super-resolution from image sequence under influence of hot-air optical turbulence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, Alaska, June 2008.

[42] J. Stone, P. H. Hu, S. P. Mills, and S. Ma. Anisoplanatic effects in finite-aperture optical systems. *Journal of the Optical Society of America A*, 11(1):347–357, Jan. 1994.

[43] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, San Francisco, CA, June 2010.

[44] M. Tahtali, A. Lambert, and D. Fraser. Self-tuning Kalman filter estimation of atmospheric warp. In *SPIE Image Reconstruction from Incomplete Data*, San Diego, CA, Aug. 2008.

[45] Y.-W. Tai, H. Du, M. S. Brown, and S. Lin. Correction of spatially varying image and video motion blur using a hybrid camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1012–1028, Jun 2010.

[46] Y.-W. Tai, P. Tan, and M. S. Brown. Richardson-lucy deblurring for scenes under a projective motion path. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1603–1618, Aug 2011.

[47] A. N. Tikhonov and V. Y. Arsenin. *Solution of Ill-Posed Problems*. Winston, 1977.

[48] P. Van Overschee and B. De Moor. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

[49] B. M. Welsh and C. S. Gardner. Effects of turbulence-induced anisoplanatism on the imaging performance of adaptive-astronomical telescopes using laser guide stars. *Journal of the Optical Society of America A*, 8(1):69–81, Jan. 1991.

[50] L. Younes. *Shapes and Diffeomorphisms*, volume 171 of *Applied Mathematical Sciences*. Springer, 2010.

[51] X. Zhu and P. Milanfar. Image reconstruction from videos distorted by atmospheric turbulence. In *SPIE Electronic Imaging*, San Jose, CA, Jan. 2010.

[52] X. Zhu and P. Milanfar. Stabilizing and deblurring atmospheric turbulence. In *Proceedings of the of the IEEE International Conference on Computational Photography (ICCP 2011)*, Carnegie Mellon University, Pittsburgh, PA, Apr. 2011.