

A linear systems approach to 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 described as the cumulative effect of a blurring kernel and a time-dependent deformation of the image domain. We introduce a statistical dynamic model for the generation of turbulence based on linear dynamical systems (LDS). 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 isoplanatic. 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 distance between the observed objects and the camera, and by other artifacts introduced by the device itself. The problem of reconstructing an image that is altered by ground-level atmospheric turbulence is 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 (any two consecutive images in the figure are 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 (see [12, 34, 45] 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 [31, 32, 36, 40]) 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 [14, 35]: 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. Both the LF and LRF approaches suffer from the limitation that it is assumed that the image sequence is extracted from short-exposure videos,



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



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

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.* [38] propose a three-stage process: (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.* [16] 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 [10, 20]. They perform image registration via Large Deformation Diffeomorphic Metric Mapping (LDDMM), introduced by Miller *et al.* [4, 30, 46] which is computationally very expensive; moreover the final results depend heavily on the choice of the parameters of the LDDMM registration. Li *et al.* [24] 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 [24] nor [21] address the issue of domain deformation and thus are only suitable when the geometric distortion is reasonably small. Tahtali *et al.* [41] 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 position of each pixel and produce a static image. However: the state space dimension of the model is so large to be untreatable; to circumvent this issue they treat each pixel location and dimension independently; by their own admission, the results depend very heavily on the initial guess of the initial value of the natural frequency at each location; they only test their technique on artificial data. Zhu and Milanfar [47] 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 [48] they replace the second step by an algorithm that yields a static

image with supposedly space-invariant blur by combining sharp patches in the registered sequence, which is followed by a final blind deconvolution step. We should mention a recent paper by Mao and Gilles [28] where they simultaneously register the 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. Tai *et al.* [25, 42] have proposed solutions to the *motion blur* problem (previously addressed e.g. in [3]), which occurs when parts of the image region (or its entirety) are affected by space-varying blur caused by either camera or object motion; 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 exception of [41], 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. It is thus natural to model the image sequence as the output of a linear, time-invariant stochastic linear dynamical system (LDS) [23]. Also, the spatial redundancy of each frame 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 approach [11] to model the image formation process as a LDS 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 a Kalman filter (KF) is used to estimate the underlying scene, which is supposed to be static; this step yields an image whose blurring is mostly isoplanatic. Applying nonlocal Total Variation (NL-TV) schemes [15, 27] to this image produces a sharp final result.

2 Problem setup

Figure 2 suggests that what is intuitively referred to as “turbulence” can be modeled, at least to a first approximation, as the combined effect of (i) a blur with an anisoplanatic point spread function and (ii) a time-dependent diffeomorphic deformation of the image domain. Such approach, also used by other authors, is somewhat artificial in that both effects 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.

Assumptions. From now on we shall assume that the camera is *static* and there is a “true” underlying scalar luminance $\rho : \Omega \rightarrow \mathbb{R}$ that we aim 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 $I : \Omega \times [0, T] \rightarrow \mathbb{R}$, for some $T > 0$. We assume that turbulence can be modeled as the concatenation of two transformations, in the specified order:

- (i) the convolution with a time-and space-invariant kernel:

$$\rho \mapsto \mathcal{K}\rho, \quad \text{with:} \quad (\mathcal{K}\rho)(x) = (k * \rho)(x) = \int_{\Omega} k(x - y)\rho(y) dy, \quad (1)$$

for some unknown function $k : \Omega \rightarrow \mathbb{R}$;

- (ii) a deformation of the domain, i.e. the composition with unknown time-dependent diffeomorphisms $\varphi(\cdot, t) : \Omega \rightarrow \Omega$, $t \in [0, 1]$, as follows:

$$\mathcal{K}\rho \mapsto I, \quad \text{with:} \quad I(x, t) = (\mathcal{K}\rho)(\varphi^{-1}(x, t)), \quad x \in \Omega, t \in [0, T]. \quad (2)$$

Note that we assume that while (i) is dispersive, (ii) is not. Both maps (i) and (ii) are linear, and so is their concatenation $\rho \mapsto \mathcal{K}\rho \mapsto I$. In fact we can write

$$I(x, t) = \int_{\Omega} K(x, y, t) \rho(y) dy, \quad x \in \Omega, t \in [0, T], \quad (3)$$

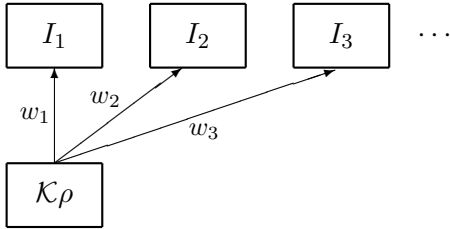


Figure 3: Relation between the original image and the data.

where the cumulative point spread function (PSF) is given by $K(x, y, t) = k(\varphi^{-1}(x, t) - y)$, $x \in \Omega$, $t \in [0, T]$, which, for any fixed t , is not isoplanatic anymore.

The model above suggests separating the image recovery problem into the estimation of the time-dependent domain deformation φ and the inference of the underlying image ρ from

$$I(\varphi(x, t)) = \int_{\Omega} k(x - y)\rho(y) dy = (\mathcal{K}\rho)(x), \quad x \in \Omega, t \in [0, T];$$

in other words, under the above assumptions the problem reduces to estimating the warping of the image sequence, followed by a deconvolution procedure.

Remark. Regarding the assumption that the camera is static, we should add that for distant scenes a known motion of the camera would not cause significant occlusions and therefore its effects could be approximated by a homography (a projective transformation) of the image. One could therefore compensate for such transformation in the computation of the above time-dependent deformation φ . Therefore our assumption is essentially made without loss of generality.

3 Dynamic models for ground-level turbulence

In order to introduce our method we shall represent the data by the image sequence $\{I_k\}$ with $I_k(x) = I(x, k\Delta t)$, where $x \in \Omega$, $k \in \mathbb{N} := \{1, 2, 3 \dots\}$ 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 (1); such transformations are precisely the time-sampled diffeomorphisms $w_k(x) := \varphi(x, k\Delta t)$, $x \in \Omega$, $k \in \mathbb{N}$, introduced in (2). In formulas:

$$I_k(x) := (\mathcal{K}\rho)(w_k^{-1}(x)) \quad \text{or simply} \quad I_k := (\mathcal{K}\rho) \circ w_k^{-1}, \quad x \in \Omega, k \in \mathbb{N} \quad (4)$$

(the symbol “:=” denotes a definition). The model is summarized in Figure 3. Note that we can write each diffeomorphic transformation $w_k : \Omega \rightarrow \Omega$ as $w_k(x) = x + W_k(x)$, $x \in \Omega$, $k \in \mathbb{N}$, where the vector field $W_k : \Omega \rightarrow \mathbb{R}^2$ 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 following sections 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 *statistical* model that relies on the dynamic texture approach introduced in [11].

3.1 An LDS-based dynamic texture algorithm

Here we review the basic dynamic texture model that will later be adapted to optical flow. Doretto *et al.* [11] introduced a statistical model to artificially synthesize time-dependent image texture for

a sequence of images $\{I_k \mid k \in \mathbb{N}\}$ by employing a dynamic model of the type:

$$\begin{cases} \xi_{k+1} = A\xi_k + q_k & q_k \sim \mathcal{N}(0, Q) \\ I_k = C\xi_k + r_k & r_k \sim \mathcal{N}(0, R) \end{cases} \quad k \in \mathbb{N}, \quad (5)$$

where $\xi_k \in \mathbb{R}^n$ is a low-dimensional state 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). When the texture has a high degree of spatiotemporal redundancy, the state ξ , which somehow summarizes the texture and its evolution, 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 $\{I_1, \dots, I_\tau\}$, 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 (5) where A , C , Q , R and the initial state ξ_1 are substituted by MAM^{-1} , CM^{-1} , MQM^{-1} , RM^{-1} and $M\xi_1$, respectively; for a given realization of the Gaussian noise terms $\{q_k\}$ and $\{r_k\}$ the two dynamical systems generate the same data sequence $\{I_k\}$). Ideally, one would want to solve the following maximum likelihood problem: given measurements $\{I_1, \dots, I_\tau\}$ (*training sequence*), with $\tau \gg n$, compute the estimates

$$(\hat{A}, \hat{C}, \hat{Q}, \hat{R}, \hat{\xi}_1) := \arg \max_{A, C, Q, R, \xi_1} p(I_1, \dots, I_\tau), \quad (6)$$

where p is the joint probability distribution of the first τ measurements that is induced by the model (5). However, given the high dimension of the measurements $\{I_k\}_{k=1}^\tau$, for computational reasons it is not practical to use the known asymptotically optimal methods for systems identification, such as [26, 39, 44]. We instead implement the technique described in [11], which is briefly summarized below; it is a fast and efficient algorithm that provides a sub-optimal solution to (6).

Summary of the algorithm. We will make the following assumptions on the matrix $C \in \mathbb{R}^{m \times n}$:

$$\text{rank}(C) = n \quad \text{and} \quad C^T C = \mathbb{I}_n \quad (7)$$

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 an essentially unique solution for the identification criterion that we consider below.

Consider the three matrices:

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

Note that by (5) 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 $\{\sigma_i \mid i = 1, \dots, \min(m, \tau)\}$, known as the *singular values* of I_1^τ , are real and non-negative. We remind the reader that the Frobenius inner product of two equally sized matrices G and H is $\langle G, H \rangle_F := \sum_{i,j} G_{ij} H_{ij} = \text{trace}(G^T H)$; 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, $(\hat{C}(\tau), \hat{\xi}_1^\tau) := \arg \min_{C, \xi_1^\tau} \|r_1^\tau\|_F = \arg \min_{C, \xi_1^\tau} \|I_1^\tau - C\xi_1^\tau\|_F$; it follows from the fixed rank approximation property [19] that

$$\hat{C}(\tau) = \text{first } n \text{ columns of } U \quad \text{and} \quad \hat{\xi}_1^\tau = \text{first } n \text{ rows of } \Sigma V.$$

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

$$\widehat{A}(\tau) = \widehat{\xi}_2^\tau (\widehat{\xi}_1^{\tau-1})^+,$$

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

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

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 [11] (which also includes Matlab code for identifying the system parameters) for more details. We shall just observe that the condition (7) makes the estimate for C uniquely determined up to a change of sign of C and ξ , and a permutation of the columns of C and the corresponding dimensions of the state space. Note, however, that it is common to order the columns of U , and therefore those of C , according to descending order of the corresponding singular values in Σ ; in addition any change of basis in the state space has in fact no influence on the output of the system. Also, the realization of the dynamical system that is obtained by imposing the orthonormality of the columns of C is “balanced”, in that it has reduced numerical sensitivity in identification and synthesis. Last, but not least, the sequence $\widehat{I}_k := \widehat{C}(\tau)\widehat{\xi}_k$, $k = 1, \dots, \tau$ 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* $\bar{I}(x) := \frac{1}{\tau} \sum_{k=1}^{\tau} I_k(x)$, $x \in \Omega$, i.e. one writes $\delta I_k(x) := I_k(x) - \bar{I}(x)$, $x \in \Omega$ and considers δI_k as the output of the dynamical system (5).

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:

$$\begin{cases} \xi_{k+1} = A\xi_k + q_k & q_k \sim \mathcal{N}(0, Q) \\ W_k = C\xi_k + r_k & r_k \sim \mathcal{N}(0, R) \end{cases} \quad k \in \mathbb{N}, \quad (8)$$

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

$$W_k = \begin{bmatrix} W_k^1 \\ W_k^2 \end{bmatrix} = \begin{bmatrix} C^1 \\ C^2 \end{bmatrix} \xi_k + r_k \quad (9)$$

where matrices C^1 and C^2 are both in $\mathbb{R}^{m \times n}$. We remind the reader that the optical flow that we are considering is calculated between a fixed image and the video sequence, as illustrated in Figure 3 (and *not* between subsequent frames in the sequence), so to model the warping from turbulence.

In principle, once the parameters have been identified, the system (8) 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_τ 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 a training sequence must

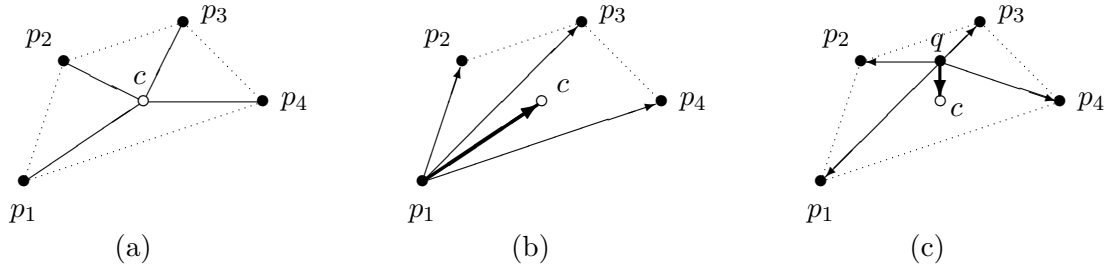


Figure 4: (a) The centroid in Euclidean space. (b) $\overrightarrow{p_1c}$ (thick) is obtained by averaging vectors $\overrightarrow{p_1p_i}$ (thin). (c) The “correction” \overrightarrow{qc} (thick) is obtained by averaging vectors $\overrightarrow{qp_i}$ (thin).

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 $\tilde{\rho}$ of the blurred true image $\mathcal{K}\rho$ and recover the sequence of optical flows from such estimate and the actual data $\{I_1, \dots, I_\tau\}$; then this estimated sequence of optical flows can be used to identify the system parameters of (8) using the dynamic texture techniques summarized in §3.1. For example, the *temporal mean* $\bar{I}(x) := \frac{1}{\tau} \sum_{k=1}^{\tau} I_k(x)$, $x \in \Omega$ could be used for $\tilde{\rho}$. 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 $\tilde{\rho}$ 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 $c = \frac{1}{N} \sum_{i=1}^N p_i$; see Figure 4(a). 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 $\overrightarrow{p_1p_i}$, $i = 2, \dots, N$ are known, then we can still compute the position of c *with respect to* p_1 as

$$\overrightarrow{p_1c} = \frac{1}{N} \sum_{i \neq 1} \overrightarrow{p_1p_i}, \quad (10)$$

as illustrated in Figure 4(b). 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 other 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 $\{I_i(x), x \in \Omega\}_{1 \leq i \leq N}$. 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” $\overrightarrow{I_1I_i}(x)$, $x \in \Omega$, $i = 1, \dots, N$; so, treating optical flow as an element of a linear space, we compute the average flow $\overrightarrow{I_1c}$ using equation (10) and then we warp I_1 via the flow $\overrightarrow{I_1c}$ to finally get the “centroid” image c of the data set.

Iterative scheme. Note however that in the case of images the procedure is *not* exact since in reality images are not points in an Euclidean space and one cannot be exactly deformed into the other by warping it via optical flow; in particular note that if turbulence causes features to be “lost” in I_1 they will not appear in c either. Therefore 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 partially correct this the following iterative scheme 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. Such procedure is similar to the one employed by Pennec [33] for computing the Fréchet (or intrinsic) mean of a distribution of points (or a probability density) 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 Horn-Schunck algorithm [22] (which has the advantage of being computationally fast;

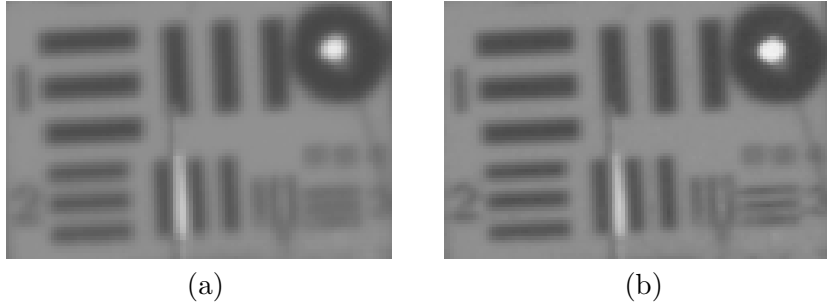


Figure 5: Comparison: (a) the temporal mean and (b) the centroid of a set of images, both computed using the first 200 samples of the sequence and, in the case of (b), 5 iterations of the scheme.

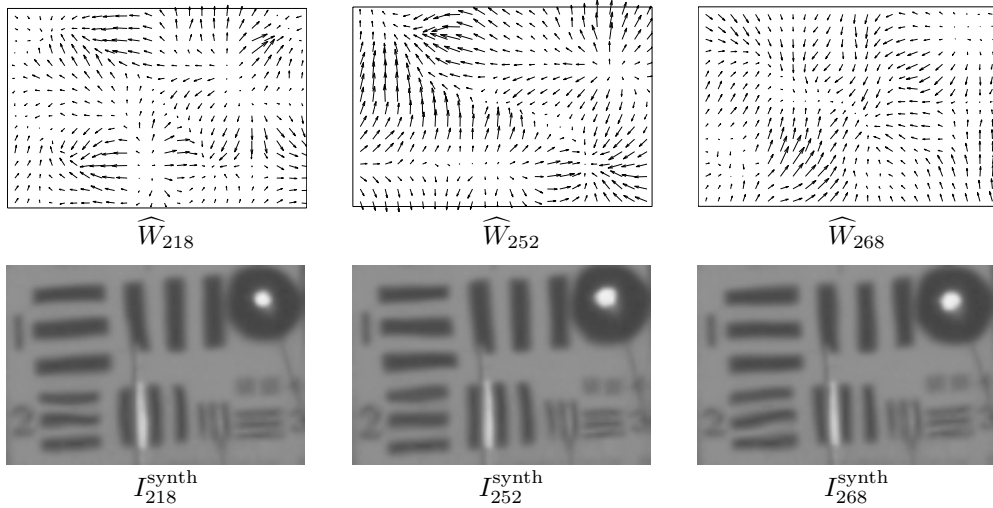


Figure 6: Three non-consecutive realizations of the synthesized flow (top), and the corresponding frames obtained by warping the centroid (bottom). For this example, $n = 50$ and $\tau = 200$.

other methods, such as Black & Anandan [5], produce sharper flows, which are not necessarily appropriate for modeling optical turbulence). On the other hand, 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 5 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 considerably sharper than the temporal mean.

Identification and simulation. After the centroid is computed, it can be used as an initial estimate $\tilde{\rho}$ of the image $\mathcal{K}\rho$ and the optical flows $\mathcal{W} := \{\tilde{W}_1, \dots, \tilde{W}_\tau\}$ between $\tilde{\rho}$ 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 (8). At this point the flow can be simulated (i.e. synthesized) 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 the dynamical system (8). 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}) can then be used to warp the centroid $\tilde{\rho}$ so to visualize the resulting (synthetic) turbulent images:

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

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

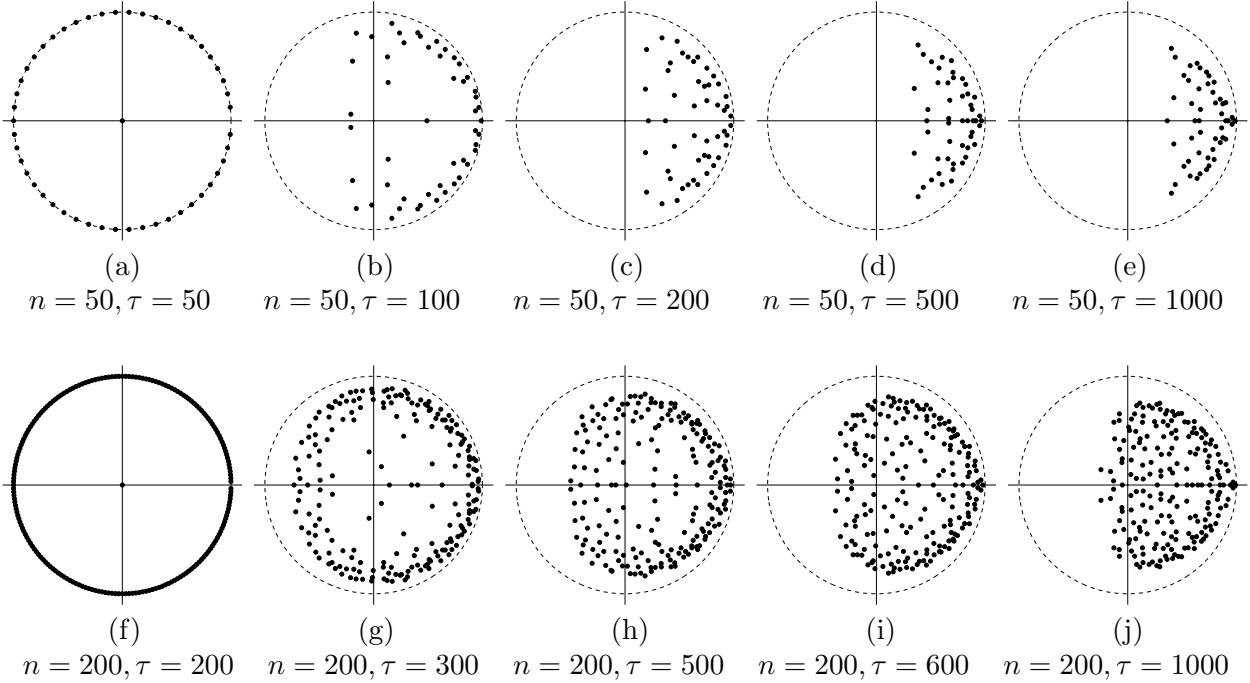


Figure 7: The eigenvalues of $\widehat{A}(\tau)$ in ten cases. The circle represents the complex curve $|z| = 1$.

Finally, Figure 7 compares the eigenvalues of the identified matrix $\widehat{A}(\tau)$ in (8) 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 (i.e. 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 $n = 50$). 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 $\tilde{\rho}$ and to identify the system parameters $\widehat{A}(\tau)$, $\widehat{C}(\tau)$, $\widehat{Q}(\tau)$, and $\widehat{\xi}_1^\tau$ for the system (8); parts (a) and (b) show, respectively, the actual frame I_{501} and the “predicted” image \widehat{I}_{501} , computed by setting $\widehat{\xi}_{501} := \widehat{A} \widehat{\xi}_{500}$, $\widehat{W}_{501} := \widehat{C} \widehat{\xi}_{501}$ and by warping the centroid $\tilde{\rho}$ via such flow. Note that since all the eigenvalues of $\widehat{A}(\tau)$ lie inside the complex unit circle it is the case that $\widehat{A}(\tau)^k \rightarrow 0$ for $k \rightarrow \infty$, so $\widehat{\xi}_{\tau+k} := \widehat{A}(\tau)^k \widehat{\xi}_\tau \simeq 0$ and $\widehat{W}_{\tau+k} := \widehat{C} \widehat{\xi}_{\tau+k} \simeq 0$ for large k , therefore

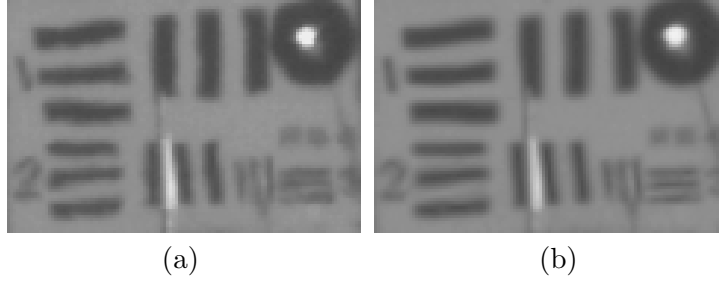


Figure 8: (a) The original frame I_{501} . (b) The one-step prediction \widehat{I}_{501} (with $n = 100$).

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 recalibrated to changing atmospheric conditions using a new training sample of images within a short time window; while we did not implement this, it could be done, for example, using the online dynamic texture estimation algorithm with exponential decaying weights described in [9].

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 simplest scheme to employ with such dynamic model is the Kalman filter (KF) [23]. 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 (8) 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:

$$\rho_{k+1} = \rho_k + s_k \quad r_k \sim \mathcal{N}(0, S) \quad (11a)$$

$$\xi_{k+1} = A\xi_k + q_k \quad q_k \sim \mathcal{N}(0, Q) \quad k \in \mathbb{N} \quad (11b)$$

$$I_k = \rho_k \circ w_k^{-1} \quad (11c)$$

where $w_k(x) = x + W_k(x)$, $x \in \Omega \subset \mathbb{R}^2$ and, by virtue of the second of equations (8), $W_k = C\xi_k$. 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. Also, s_k and q_k are independent, white, Gaussian additive noise terms, and it is the case that $Q := \widehat{Q}(\tau)$ from §3.2. Note that the additive noise s_k in (11a) will allow the estimate of ρ to drift away from the value it is initially assigned.

We call (11a) and (11b) *state equations* and (11c) *measurement equation*; the joint state sequence $\{(\rho_k, \xi_k)\}_{k \in \mathbb{N}}$ is hidden whereas all we measure is the data sequence $\{I_k\}_{k \in \mathbb{N}}$. As we shall soon see in detail we cannot apply directly the equations of the Kalman filter for estimating the hidden state. In fact on the right-hand side of the measurement equation (11c) we have both the image ρ_k and the flow w_k (which is a function of the state ξ_k via $W_k = C\xi_k$), but the equation is *not* linear in the *joint* state (ρ_k, ξ_k) . In fact it is linear in ρ_k (for fixed ξ_k) and not in the flow. So we will

have to apply an *alternate estimation* (AE) procedure at each measurement update, as explained in detail later. We define the estimates/error covariance matrix pairs:

$$\begin{aligned}\widehat{\rho}_{j|k} &:= \mathbb{E}[\rho_j | I_1 \dots, I_k], & \Sigma_{j|k} &:= \text{Var}[\rho_j | I_1 \dots, I_k], \\ \widehat{\xi}_{j|k} &:= \mathbb{E}[\xi_j | I_1 \dots, I_k], & P_{j|k} &:= \text{Var}[\xi_j | I_1 \dots, I_k].\end{aligned}$$

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 $\widehat{W}_{j|k} := \mathbb{E}[W_j | I_1 \dots, I_k]$ and the corresponding error covariance matrix $\Theta_{j|k} := \text{Var}[W_j | I_1 \dots, I_k]$. 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:

$$\widehat{\rho}_{k+1|k} = \widehat{\rho}_{k|k} \tag{12a}$$

$$\widehat{\xi}_{k+1|k} = A\widehat{\xi}_{k|k} \tag{12b}$$

$$\Sigma_{k+1|k} = \Sigma_{k|k} + S \tag{12c}$$

$$P_{k+1|k} = AP_{k|k}A^T + Q \tag{12d}$$

where $S = \mathbb{E}[s_k s_k^T]$ and $Q = \mathbb{E}[q_k q_k^T]$ are the covariance matrices introduced in (11a) and (11b).

Measurement Update (MU) equations for ρ . 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 (11c) as:

$$I_k \circ w_k = \rho_k + \zeta_k \tag{13}$$

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 issue below) the measurement update equations for ρ are simply:

$$\widehat{\rho}_{k+1|k+1} = \widehat{\rho}_{k+1|k} + L_{k+1}(I_{k+1} \circ \widehat{w}_{k+1|k+1} - \widehat{\rho}_{k+1|k}), \tag{14}$$

$$\Sigma_{k+1|k+1} = (I - L_{k+1})\Sigma_{k+1|k},$$

$$\text{where } L_{k+1} = \Sigma_{k+1|k}(\Sigma_{k+1|k} + Z)^{-1}$$

is the Kalman gain. Note that in equation (14) we need the estimate $\widehat{w}_{k+1|k+1}$, i.e. $\widehat{W}_{k+1|k+1} := \mathbb{E}[W_{k+1} | I_1 \dots, I_{k+1}] = C\widehat{\xi}_{k+1|k+1}$, 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 (12b); in the next step (measurement update for ξ) we will use the value of $\widehat{\rho}_{k+1|k+1}$ from (14) to compute $\widehat{\xi}_{k+1|k+1}$, and we shall then iterate the process.

Measurement Update (MU) equations for ξ . We now assume that $\widehat{\rho}_{k+1|k+1}$, computed with the procedure described above, is known. Whence the most natural way of performing the measurement update for ξ is to consider the following linear measurement equation instead of (11c):

$$W_k^{\text{meas}} = C\xi_k + r_k, \quad r_k \sim \mathcal{N}(0, R), \tag{15}$$

which is precisely the second of equations (8), 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 again be computed via a modern implementation of the Horn-Schunck algorithm [22]. So the measurement update equations for ξ 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

$$\begin{aligned}M_{k+1} &= P_{k+1|k}C^T(CP_{k+1|k}C^T + R)^{-1}, \\ \text{or: } M_{k+1} &= (C^TR^{-1}C + P_{k+1|k}^{-1})^{-1}C^TR^{-1}\end{aligned}\tag{16}$$

where we have applied the Matrix Inversion Lemma [23] (under the assumption that $P_{k+1|k} > 0$). Note that (17) is more convenient to use than (16) because (17) involves the inversion of two $n \times n$ matrix, while (16) involves the inversion of an $m \times m$ matrix, with $m = m_1m_2 \gg n$. The inversion of the $m \times m$ matrix R is performed, offline, only once; however, for computational purposes, it may be convenient to use instead a diagonal matrix R with equal diagonal elements.

Remark (Measurement Update via an Alternating Estimation (AE) scheme). As noted before, the measurement equation (11c) is not linear in the joint state (ρ_k, ξ_k) therefore we employ an alternate estimation scheme. However, as we said above in order to compute $\widehat{\xi}_{k+1|k+1}$ in principle one needs $\widehat{\rho}_{k+1|k+1}$, and vice versa. A reasonable procedure is to alternate between the two measurement updates for ρ (for ξ fixed) and ξ (for ρ fixed), by using $\widehat{\xi}_{k+1|k}$ instead of $\widehat{\xi}_{k+1|k+1}$ at the first step. In detail, for a fixed k :

1. Perform the TU (time update) by equations (12a)–(12d).
2. Use $\widehat{\xi}_{k+1|k}$ instead of $\widehat{\xi}_{k+1|k+1}$ in the MU equations for ρ to compute $\widehat{\rho}_{k+1|k+1}$ and $\Sigma_{k+1|k+1}$.
3. Use $\widehat{\rho}_{k+1|k+1}$ in the MU equations for ξ to compute $\widehat{\xi}_{k+1|k+1}$ and $P_{k+1|k+1}$.
4. Use $\widehat{\xi}_{k+1|k+1}$ in the MU equations for ρ to compute $\widehat{\rho}_{k+1|k+1}$ and $\Sigma_{k+1|k+1}$.
5. Go to step **3** and iterate.

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 $m = m_1m_2 \gg n$) are treated as scalars for computational reasons.

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 parameters of the dynamical system (8). 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 k th 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. 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 blurring kernel; in this section we briefly describe the problem and summarize our method of choice. Given an

unknown image $u : \Omega \rightarrow \mathbb{R}$ and an convolution kernel $k : \Omega \rightarrow \mathbb{R}$ (also referred to as point spread function, or PSF) the well-known *deconvolution problem* is the one of recovering u from the data:

$$f := k * u + n \quad (18)$$

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 the higher frequencies of u . So in order to make the solution to the problem unique one needs to restrict the search of u to a smaller function space. One way to proceed is to define the solution \hat{u} of the deconvolution problem as

$$\hat{u} = \arg \min_u \left\{ \int_{\Omega} (f - k * u)^2 dx + \mu \int_{\Omega} U(u, \nabla u) dx \right\}, \quad (19)$$

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. One can choose $U(u, \nabla u) = u^2$, i.e. impose an L^2 regularization term [43]; the 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. Instead, one can use an H^1 semi-norm regularization term [1], i.e. choose $U(u, \nabla u) = \|\nabla u\|^2$; in this case, which is also referred to as *Wiener filtering*, the Euler-Lagrange equation is the PDE: $k^* * (k * u - f) - \mu \Delta u = 0$, where Δu is the Laplacian of u . Both methods thus lead to linear equations and it is possible to employ the FFT to solve them in *one step* in the frequency domain. However, to compensate for the noise a large regularization parameter μ must be used, which causes the edges of \hat{u} to be smeared out.

To circumvent this problem Rudin, Osher and Fatemi [37] 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

$$\frac{\partial u}{\partial t} = \lambda k^* * (f - k * u) + \mu \operatorname{div} \left(\frac{\nabla u}{\|\nabla u\|} \right), \quad (20)$$

with Neumann boundary condition $\frac{\partial u}{\partial \mathbf{n}} = 0$ on $\partial\Omega$. Methods for implementing (20) are described, for example, in [8, 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 [15, 27]. This is particularly effective when image structures (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 (18), first one considers a nonnegative, symmetric *weight* function $w_f(x, y)$, $(x, y) \in \Omega^2$, with the property that $w_f(x, y) \simeq 1$ when the patches around locations x and y are similar and $w_f(x, y) \simeq 0$ otherwise. For example one can choose (as in [6, 7]):

$$w_f(x, y) := \exp \left\{ - \frac{(G_a * |f(x + \cdot) - f(y + \cdot)|)(0)}{h^2} \right\}, \quad (21)$$

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 (18). 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:

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

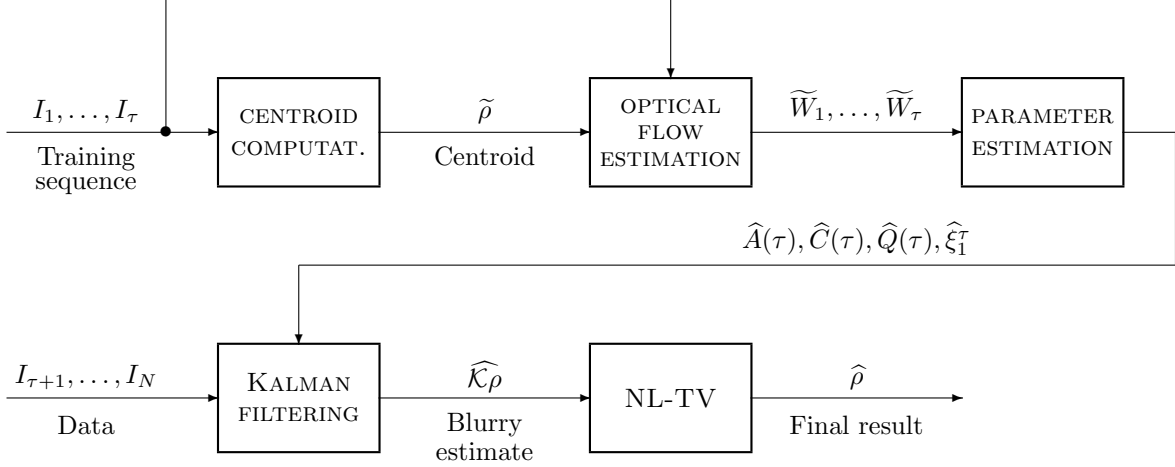


Figure 9: Flow chart of the algorithm.

Also, given a function $v \in L^2(\Omega^2, \mathbb{R})$ we can define its *nonlocal* divergence $\text{div}_w v \in L^2(\Omega, \mathbb{R})$ as the dual of the nonlocal gradient with respect to the L^2 inner product, that is, by definition: $\langle \text{div}_w v, u \rangle_{L^2(\Omega, \mathbb{R})} = -\langle v, \nabla_w u \rangle_{L^2(\Omega^2, \mathbb{R})}$; it is the case [15] that this definition leads to the expression

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

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

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

One can show [15] 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

$$u_t = \lambda k^* * (f - k * u) - \mu L[u], \quad (22)$$

$$\text{where } L[u] := -\text{div}_w \left(\frac{\nabla_w u}{|\nabla_w u|} \right) = -\int_{\Omega} [u(y) - u(x)] \left(\frac{1}{|\nabla_w u|(x)} + \frac{1}{|\nabla_w u|(y)} \right) w(x, y) dy,$$

which is obtained by computing the first variation of $J[u]$. Note that the PDE (22) is formally similar to (20), except that the divergence term is substituted by its nonlocal version. Note that the regularizing term J and the corresponding gradient term L also depend on the data f . We used precisely NL-TV deconvolution to sharpen the blurry image $\hat{\rho}_{N|N}$ provided by the Kalman filter. More implementation details, as well as our final results, are discussed in the next section.

6 Numerical results

Our image recovery algorithm is summarized in Figure 9. First of all a training sequence I_1, \dots, I_{τ} is used for computing the centroid. The centroid and the training sequence itself are employed to compute an estimate of the sequence of motion fields that generate the data, i.e. the optical flows $\tilde{W}_1, \dots, \tilde{W}_{\tau}$ between the centroid $\tilde{\rho}$ and the data I_1, \dots, I_{τ} . Following this, such sequence

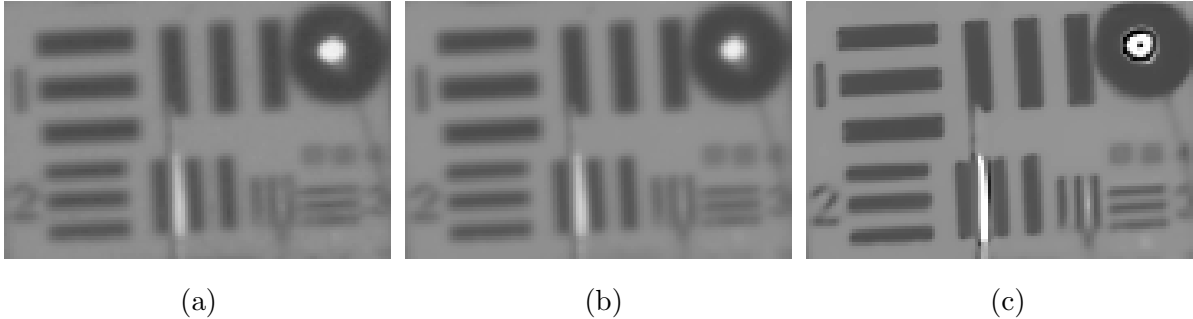


Figure 10: Comparison between: (a) the centroid of the first 100 images; (b) the blurry image produced by 200 iterations of Kalman filtering (with $n = 50$); and (c) the image resulting from NL-TV deconvolution.

of motion fields is used to identify the parameters of the dynamical system (8), which models the time evolution of turbulence, by following the procedure of §3.2. A Kalman filter is then applied to the “global” dynamical system (11), with the parameters identified at the previous step and using the centroid $\tilde{\rho}$ and $\hat{\xi}_\tau$ as initial estimates for ρ and ξ . For the reasons discussed in §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, an NL-TV deconvolution scheme is employed for deblurring. 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 hand-picked standard deviation.

Figure 10 shows the result of our algorithm with the intermediate steps, i.e. the centroid and the blurry image produced by the Kalman filtering procedure. We used a training sequence of $\tau = 200$ images $\{I_1, \dots, I_\tau\}$, fixed the dimension of the dynamical system (8) at $n = 50$ and used a data sequence of 200 images to perform the Kalman filtering. The standard deviation of the Gaussian blurring kernel used in the NL-TV algorithm was hand-picked at 1.5, as it yielded the crispest result. The final result is better than then one obtained by simply applying NL-TV directly to the temporal mean, or the centroid: this is especially apparent when analyzing the smaller features of the recovered images. In Figure 11 the first image lacks details and sharpness. The second one has excessive contrast between the horizontal bars, which is an artifact that typically arises in the application of a deblurring algorithm to an image with anisoplanatic (i.e. non-space-invariant) blur. This does not occur in the third image, indicating that the Kalman filter produces an image with blur that is essentially isoplanatic.

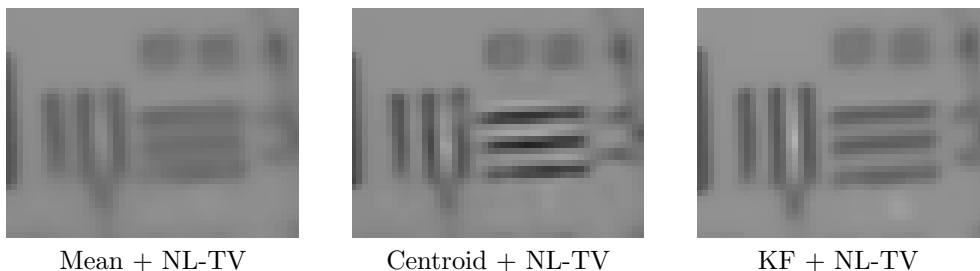


Figure 11: Comparison of the images resulting from applying NL-TV to the temporal mean (of 100 samples), the centroid (also computed with 100 samples), and the Kalman filter output.

Figure 12 illustrates the raw data, intermediate steps and final results yielded by our scheme for three image sequences of the same scene taken at different times of the day, therefore with

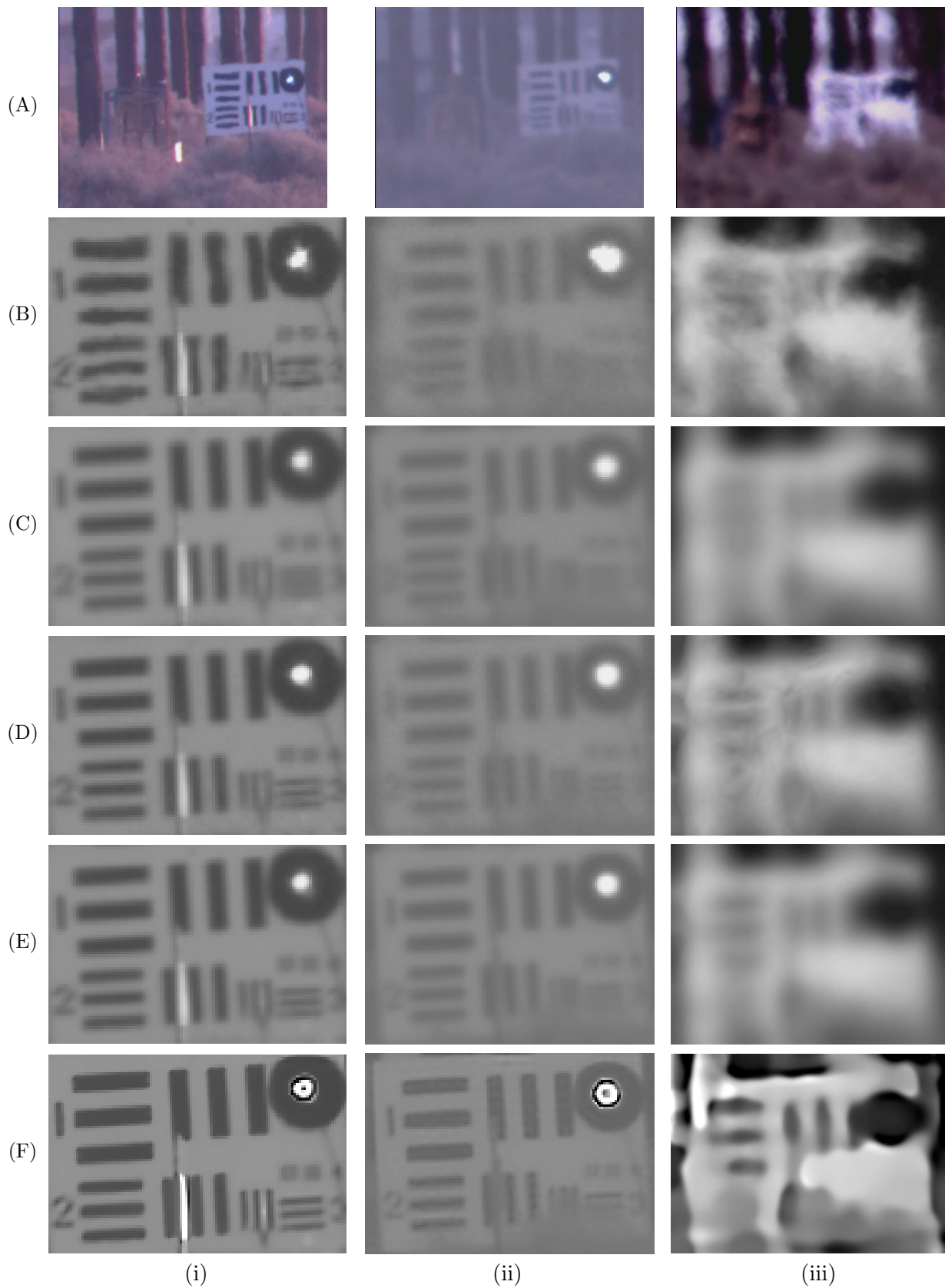


Figure 12: From raw data to recovered images for three sequences: (A) a complete frame in the original sequence; (B) detail; (C) temporal mean (never used); (D) centroid; (E) Kalman Filter output; (F) NL-TV output.

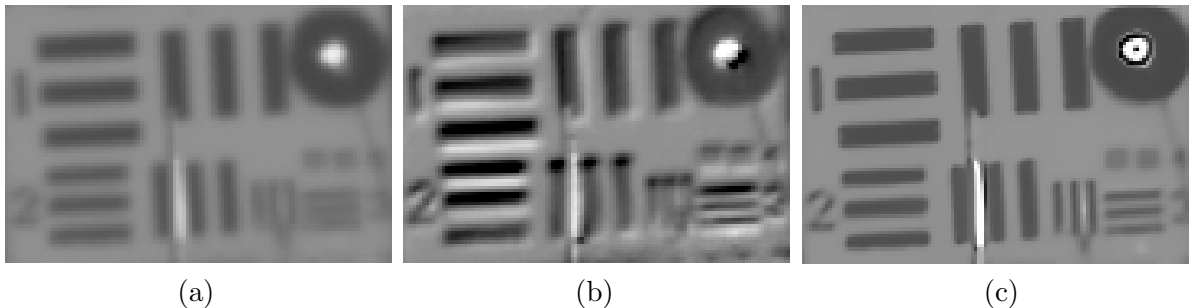


Figure 13: Comparison between the final results yielded by: (a) the Lucky region fusion (LRF) approach; (b) the PCA method; and (c) our algorithm.

different temperature, pressure, humidity, wind, and lighting conditions. Sequence (i), on the left, was taken in the morning and in fact is the one that we have considered so far in this paper. Sequence (ii) is taken in the evening and is characterized by an intermediate level of turbulence, with a higher blur, as apparent from Figure (B)-(ii), lower contrast, and lower luminance; note how the final result is still relatively crisp. Image sequence (iii) was taken in the afternoon and is characterized by turbulence conditions that are so extreme that the scene is hardly visible: parts of it are in fact completely saturated by light. Note, in this example, how the image (E)-(iii) yielded by the Kalman filter does not have the artifacts that are present in the centroid (D)-(iii), while having similar geometrical structure. It is also much sharper than the temporal mean (C)-(iii). As expected the results are worse than in the previous two experiments, but some parts of the board (such as the black circle and the thicker bars) are clearly visible after applying our recovery scheme. As we describe next we used the same hand-picked parameters in all three experiments, except for the standard deviation of the Gaussian kernel in NL-TV deconvolution.

Finally, in Figure 13 we compare the results yielded by two state of the art techniques and our own method. The first image is obtained by applying the Lucky Region Fusion (LRF) method [2] to the first 100 frames of the image sequence that we have treated throughout this paper (column (i) of Figure 12), and we can see that while the geometry is reconstructed well the final result is still considerably blurry. The second image in Figure 13 is yielded by the Principal Component Analysis (PCA) technique described in [24], once again applied to the first 100 images of the same sequence: this method also reconstructs the geometry of the underlying scene correctly, and does a better job at deblurring than LRF does. However our method (albeit considerably slower than the first two) yields a result, shown on the right, that is considerably sharper.

Free parameters, and computational complexity. Throughout the algorithm there is a number of hand-tuned parameters which we summarize here, pointing out those to which our approach is most sensitive.

- (1) **Centroid computation.** The parameters to choose are: the smoothness parameter to be used in the Horn-Schunck algorithm¹ for optical flow recovery, which we call α ; the number τ of images I_1, \dots, I_τ to use be used in the computation of the flow (10); the number M of cycles in the iterative scheme used to compute the centroid. In our experiments we chose $\alpha = 80$, $\tau = 100$ and $M = 5$ (as long as $M \geq 4$ the centroid is fairly robust with respect to the choice of such parameters). An important choice is the reference image with respect to which the centroid is computed (i.e. I_1 in Figure 4); in fact if turbulence has eliminated certain features from this particular frame, these will also be absent from the centroid (however, the Kalman filter scheme will later compensate for such deficiencies by using the rest of the data). We used bicubic interpolation to warp the images via the computed flows.

¹We remind the reader that the Horn-Schunck method minimizes a data fidelity term for the flow plus (as regularizing term) the H^1 -seminorm of the flow itself times a smoothing parameter; see [22] for details.

- (2) **Computation of the training sequence** $\{\widetilde{W}_k\}$. This is the sequence of the optical flows between the centroid and the data set, to be used for parameter identification in the linear system (8). One has to choose the length τ of the training sequence and, once again, the smoothing parameter α for the computation of such optical flows. In our experiments τ is in the 100-400 range, and $\alpha = 80$ as above. Again, the results are not very sensitive to the choice of such parameters.
- (3) **LDS parameters identification**. For this task one chooses the dimension n of the internal state ξ of the system (8), with $n < \tau$. In our experiments n is in the 50-200 range.
- (4) **Kalman Filter**. The two parameters that have to be chosen for the Kalman Filter are Z and R , the noise variance values in the measurement equations (13) (for ρ) and (15) (for ξ), respectively. In our experiments we have chosen $Z = 5000$ and $R = 50$, so to allow ρ to evolve from its initial value (which is the centroid). The KF scheme also computes, at each step, one or more optical flow estimates between pairs of images, depending on how many alternative estimation (AE) cycles we decide to run at each step (in our experiments, it is two): the smoothness parameter α is fixed at 80, as above.
- (5) **NLTV**. First of all, the weight w_f defined in (21) must be computed from the Kalman filter output. This is normally done by pre-processing the data with the Wiener filter, i.e. (19) with $U(u, \nabla u) = \|\nabla u\|^2$, in order to enhance edges and thus impose a structure to the NL-TV estimate (through the weight function w_f): the parameter μ in (19) is computed automatically (so to make the H^1 seminorms of f and \hat{u} the same). Two parameters that are hand-picked are the size of the patch and the size of the search window for the construction of w_f . Last, the weights in the NL-TV functional must be chosen, as well as the variance of the Gaussian kernel used in the deconvolution. Of all the parameters listed here the latter is by far the most important one, as it is normally the case in deconvolution problems.

As far as computational complexity is concerned, most parts of the algorithm run in real-time except for those involving the estimation of optical flow: for two 90×130 images, the such computation takes about 3.7 seconds using Matlab on a MacBook equipped with a 2.4 GHz Intel Core 2 Duo processor. This makes the computations in (1), (2) and (4) the most burdensome parts of the scheme. Another computationally expensive part of the algorithm is the calculation of the weight function employed by the NL-TV scheme, though this has to be done only once.

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 temperature, pressure, humidity, wind and lighting conditions, together 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 estimate of the original image.

We should point out that there are situations when the proposed method does **not** work: namely, those where a linear system of the type (8) 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.

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: finding an algorithm for the centroid computation that is robust to the choice of the reference image; incorporating the deblurring step into the Kalman filtering scheme; considering ways to automate the choice of what now are hand-picked parameters; and exploring the advantages of choosing different deblurring kernels, such as those known from the optics literature (see [13, 17, 29] 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 (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 the 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] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. In *In Proceedings of the Fourth European Conference on Computer Vision*, pages 573–582. Springer-Verlag, 1996.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.

- [8] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20:89–97, Jan 2004.
- [9] A. B. Chan, V. Mahadevan, and N. Vasconcelos. Generalized Stauffer–Grimson background subtraction for dynamic scenes. *Machine Vision and Applications*, 22(5):751–766, Sept. 2011.
- [10] T. F. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE Transactions on Image Processing*, 7(3):370–375, Mar. 1998.
- [11] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.
- [12] D. L. Fried. Statistics of a geometric representation of wavefront distortion. *Journal of the Optical Society of America*, 55(11):1427–1435, Nov. 1965.
- [13] 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.
- [14] 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.
- [15] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling and Simulation*, 7(3):1005–1028, 2008.
- [16] J. Gilles, T. Dagober, 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.
- [17] J. Gilles and S. Osher. Fried deconvolution. CAM Technical Report 11-62, UCLA Department of Mathematics, Dec. 2011.
- [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] P. R. Kumar and P. Varaiya. *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Prentice Hall, Englewood Cliffs, New Jersey, 1986.
- [24] 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.
- [25] 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.

- [26] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Englewood Cliffs, NJ, 2nd edition, 1999.
- [27] Y. Lou, X. Zhang, S. Osher, and A. Bertozzi. Image recovery via nonlocal operators. *Journal of Scientific Computing*, 2(42), Feb. 2010.
- [28] Y. Mao and J. Gilles. Non-rigid geometric distortions corrections – application to atmospheric turbulence stabilization. *Inverse Problems and Imaging*, 6(3), August 2012. To appear.
- [29] S. Metari and F. Deschênes. A new convolution kernel for atmospheric point spread function applied to computer vision. In *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV 2007)*, Rio de Janeiro, Brazil, Oct. 2007.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] X. Pennec. *Probabilities and Statistics on Riemannian Manifolds: a Geometric Approach*. Research Report 5093, INRIA, January 2004.
- [34] F. Roddier. *Adaptive Optics in Astronomy*. Cambridge University Press, 2004.
- [35] 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.
- [36] M. C. Roggemann and B. M. Welsh. *Imaging through Turbulence*. CRC Press, 1996.
- [37] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.
- [38] 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.
- [39] R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3(4):253–264, 1982.
- [40] 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.
- [41] 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.
- [42] 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.
- [43] A. N. Tikhonov and V. Y. Arsenin. *Solution of Ill-Posed Problems*. Winston, 1977.
- [44] P. Van Overschee and B. De Moor. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

- [45] 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.
- [46] L. Younes. *Shapes and Diffeomorphisms*, volume 171 of *Applied Mathematical Sciences*. Springer, 2010.
- [47] X. Zhu and P. Milanfar. Image reconstruction from videos distorted by atmospheric turbulence. In *SPIE Electronic Imaging*, San Jose, CA, Jan. 2010.
- [48] 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.