Remote Sens. **2012**, *4*, 2329-2355; doi:10.3390/rs4082329

OPEN ACCESS

Remote Sensing

ISSN 2072-4292

www.mdpi.com/journal/remotesensing

Article

Four Methods for LIDAR Retrieval of Microscale Wind Fields

Allen Q. Howard, Jr. [1,2,*] **and Thomas Naini** [2]

[1] Faculdade de Geofisíca, Instituto de Geociênces, Universidade Federal do Pará, Rua Augusto Correa, 01-Guamá, Belem-PA, 66075-110, Brazil

[2] Department of Physics, Utah State University, Logan, UT 84322, USA; E-Mail: tomnaini@gmail.com

* Author to whom correspondence should be addressed; E-Mail: terragraf@gmail.com;
Tel.: +55-435-757-3206; Fax: +55-435-797-2492.

Received: 25 June 2012; in revised form: 25 July 2012 / Accepted: 27 July 2012 / Published: 8 August 2012

Abstract: This paper evaluates four wind retrieval methods for micro-scale meteorology applications with volume and time resolution in the order of 30 m^3 and 5 s. Wind field vectors are estimated using sequential time-lapse volume images of aerosol density fluctuations. Suitably designed mono-static scanning backscatter LIDAR systems, which are sensitive to atmospheric density aerosol fluctuations, are expected to be ideal for this purpose. An important application is wind farm siting and evaluation. In this case, it is necessary to look at the complicated region between the earth's surface and the boundary layer, where wind can be turbulent and fractal scaling from millimeter to kilometer. The methods are demonstrated using first a simple randomized moving hard target, and then with a physics based stochastic space-time dynamic turbulence model. In the latter case the actual vector wind field is known, allowing complete space-time error analysis. Two of the methods, the semblance method and the spatio-temporal method, are found to be most suitable for wind field estimation.

Keywords: LIDAR; 3D; vector wind fields; spatio-temporal and semblance methods; fluid flow models; retrievals

1. Introduction

Space-time volume profiling is expected to be an important tool for designing optimum wind farm parameters for wind power generation, facilitating the next-generation tools for large area 3D wind modeling over complex terrains, improving wind turbine performance, increasing turbine life, and reducing turbine operating and maintenance costs. Presently, anemometers are commonly used to sample wind fields at several selected points or regions. Newer large turbines have hub heights larger than 130 m. Anemometer masts need to be at hub height while masts higher than 100 m are problematic [1]. For this reason our interest is in volume wind methods with voxel element volumes in the order of 30 m^3. Our work here focuses on software processing from a suitable volume scanning backscatter LIDAR system. Recently, a commercial conical scanning vertical pointing aerosol LIDAR system specifically designed for the wind industry has been introduced [2]. We plan to discuss ideas for a volume prototype LIDAR system elsewhere. Our modeling study does not include measurement errors, because they are yet unknown and system dependent.

Measurement of wind fields using elastic backscatter LIDAR with short pulse and angular scanning capability has better time and space resolution than radar or sodar systems [3]. Non-Doppler or direct detection scanning LIDARs are intrinsically more simple as they do not need a coherent detection transceiver. More importantly, backscatter LIDAR can measure equally well all three components of the wind field. A Doppler system measures only the radial component of the velocity field. Another point in favor of non-Doppler systems is that they use smaller wavelengths. Typically Doppler systems use 2 or 10 μm wavelengths. Because the aerosol particles distribution functions are predominately in the 0.5 to 5 μm range, the associated backscattering cross sections at the Doppler wavelengths are relatively small. In the last decade, commercial high-power eye-safe lasers operating at 1.5 μm have become available, making it possible to build scanning, eye-safe, backscatter LIDAR systems with sufficient signal-to-noise for volume scans in the order of $1\,\mathrm{km}^3$ with ranges of a few kilometers. This opens the door for possible non-Doppler, volumetric three-component LIDAR wind field measurements ideally suited for wind farm siting and evaluation. Most of the required components for such systems are commercially available. Wilkerson [4] is one example method for validation of such a LIDAR retrieval system. Comparisons with tracking systems are not comprehensive, nor are comparisons with anemometers, but they are a place to start and would lead to a better understanding of system performance.

Features in the LIDAR backscatter patterns are caused by lower atmospheric aerosol particle loading. Particles with diameters D in the order of a wavelength of the incident radiation are resonant scatterers. This corresponds to D on the interval of approximately $[0.5, 5.0]\,\mu$m. Because lower atmosphere omni-present aerosol particles have little associated inertia and their characteristic Stokes times τ_s [5] for responding to applied forces due to wind fields are in the order of milliseconds, tracking aerosol pattern movement does correspond accurately with wind field motion.

Motion inferred from time-lapse LIDAR imagery has been developed by many authors. An important early example is by Eloranta [6]. In contrast, a paper [7] includes a review of the technological advances in the intervening 40 years. The reference [1] contains a comparison of sodar, LIDAR, and anemometer wind measurement technologies. Hasager *et al.* [8] determine meso-scale wind fields over the sea by analyzing associated ocean wave motion using satellite C-band synthetic aperture radar (SAR). The results compare favorably with mast based anemometers. Key papers on the development of LIDAR measurement and retrieval of wind fields include [1,9–18].

Here, because of the interdisciplinary nature of this paper, it seems appropriate to include both a summary and comparison of four alternative vector wind field retrieval methods. All methods compare successive time lapse imagery. The methods are (a) cross correlation method (CCM), (b) semblance method (SM), (c) translation phase shift method (TPSM), and (d) a spatio-temporal method (STM). The first three methods use a combination of segmentation and Fourier transform (FFT) processing. STM uses smaller neighborhood processing and therefore applies directly in the space and time domain. Care is taken to make all the methods numerically efficient. CCM is defined in one dimension, but has obvious extension to two and three spatial dimensions using multi-dimensional FFT's. Vector field examples in two dimensions using synthetic time lapse target imagery are used to illustrate the methods. To compare and evaluate these four methods, we have implemented a version of Stam's [19,20] realistic physics based dynamic turbulent wind field model. The four dimensional space-time model uses 4D FFT's and is thus computationally efficient.

Once the dynamic 3D vector field has been estimated, it remains a non-trivial task to visualize the dynamical results. Indeed, computer visualization is intrinsically two-dimensional; the three-dimensional vector field has seven variables, four independent (x, y, z, t) and three dependent v_x, v_y, v_z. Fortunately, visualization of dynamic vector fields is an area of active research; see for example [21,22]. The time dimension naturally leads to animation graphics using for example an avi file format. Then, because wind fields are usually primarily horizontal near the earth's surface, cuts of constant altitude reduce the display to five dimensions. The x, y and component v_x, v_y horizontal vectors, *i.e.*, $\mathbf{v}_h = v_x\hat{\mathbf{x}} + v_y\hat{\mathbf{y}}$, are displayed as arrows at the grid point (x_n, y_m) with the length of the arrow proportional to the horizontal wind speed $v_h = (v_x^2 + v_y^2)^{1/2}$. Finally, the color of the vector $\mathbf{v}_h$ is chosen from a color pallet with for example the red end of the pallet for maximum positive ratio v_z/v_h and the blue end for maximum negative ratio v_z/v_h.

Dense rectangular arrays of length-modulated and color-coded arrows are not eye friendly. Streamlines are more intuitive and yield a less cluttered map. Completing the visualization are streamlets, moving along the streamlines with speed proportional to v_h and color as before encoding the ratio v_z/v_h.

An important processing detail relates to the aerosol backscatter data. Signal and image processing are most directly accomplished in rectangular coordinates. Each volume of spherical coordinate data thus needs to be interpolated onto a rectangular voxel grid. Note $30\,\mathrm{m}^3$ resolution over volumes in the order of $1\,\mathrm{km}^3$ could result in a computational bottleneck. Since the scan is doubly periodic, volume interpolation is done every period. The period is in the order of several seconds. An efficient strategy is to pre-compute locations for a given number of nearest neighbor pointers, say $\mathcal{L} = 3, 5, 7$. These are then used to interpolate between the spherical grid neighboring points $\boldsymbol{\zeta_n}$, enumerated as $n = 1, 2, \cdots, N$ to the center of each voxel $\mathbf{x}_m\, , m = 1, 2, \cdots, M$ in the Cartesian grid. The vector subscripts $\ell(m, j)$, $j = 1, 2, \cdots, \mathcal{L}$ defining the $\mathcal{L}$ spherical nearest neighbors of each Cartesian coordinate point $\mathbf{x}_m$ are defined as $[n(\ell(m, 1)), n(\ell(m, 2)), \cdots, n(\ell(m, \mathcal{L}))]$. The matrix of vector subscripts $\ell(m, j)$ of size $[M, \mathcal{L}]$ is saved along with the derived weights $\mathcal{W}_{m\,n(\ell(m,j))}$, $j = 1, 2, \cdots, \mathcal{L}$ that are chosen to be inversely proportional to distance $\mathcal{D}_{nm} = |\mathbf{x}_m - \boldsymbol{\zeta_n}\,|$ between voxel center and spherical grid neighboring points. The weights are normalized such that

$$\sum_{j=1}^{\mathcal{L}} \mathcal{W}_{m\,n(\ell(m,j))} = 1$$

Then for each data cycle, the spherical scan can be rapidly interpolated onto the Cartesian grid as weighted averages. More explicitly let $\rho_s(n)\, n = 1, 2, \cdots, N$ be the measured 3D aerosol density fluctuation from the scanning LIDAR backscatter returns in spherical coordinates and let $\rho_c(m)\, m = 1, 2, \cdots, M$ be the aerosol density fluctuation at the center of the m^{th} Cartesian grid point. Then the interpolation is

$$\rho_c(m) = \sum_{j=1}^{\mathcal{L}} \mathcal{W}_{m\, n(\ell(m,j))}\, \rho_s(n(\ell(m,j)))$$

Simulations of this 3D interpolation scheme at 100 m^3 resolution over a 1 km^3 scan volume, corresponding to 1.0×10^7 voxels, takes 0.6 s for $\mathcal{L} = 5$ on a quad Intel processor in 64 bit Matlab®.

The next eight sections summarize the four alternative and complementary methods for estimating vector fields from time lapse imagery. Then we discuss some image processing and filtering of the data that proceeds wind field estimation. This is followed by a derivation of the dynamic turbulent wind field model and a method for advecting aerosol particles in the resulting wind field. The paper concludes with comparisons of the four wind field estimation methods using the modeled dynamic wind field data.

2. Cross Correlation Method (CCM)

In this and the following three sections, methods are stated for the most simple one-dimensional case. However, the results easily generalize to two and three-dimensions. Details of the implementation of these methods can be found in [23]. The normalized cross-correlation function $C_{\mathrm{rs}}(x)$ is defined as

$$C_{\mathrm{fg}}(x) = \frac{\mathrm{Re} \int_{-\infty}^{\infty} f(x+x')\, g^*(x')\, dx'}{\sqrt{\int_{-\infty}^{\infty} |f(x')|^2\, dx' \int_{-\infty}^{\infty} |g(x')|^2\, dx'}} \tag{1}$$

It can be shown that

$$-1 \leq C_{\mathrm{fg}}(x) \leq 1 \tag{2}$$

When $C_{\mathrm{fg}}(x) = 1$, the signals have perfect correlation for offset x.

3. Computation

Computation of $C_{\mathrm{fg}}(x)$ is usually performed via Fourier transforms where

$$F(K) = \mathcal{F}(f(x)) = \int_{\infty}^{\infty} f(x)\, e^{-iKx}\, dx \tag{3}$$

and

$$f(x) = \mathcal{F}^{-1}(F(K)) = \int_{\infty}^{\infty} F(K)\, e^{iKx}\, \frac{dK}{2\pi} \tag{4}$$

If signals are normalized such that

$$\begin{array}{c} \int_{-\infty}^{\infty} |f(x')|^2\, dx' = 1 \\ \int_{-\infty}^{\infty} |g(x')|^2\, dx' = 1 \end{array} \tag{5}$$

Then as in [24]

$$C_{\mathrm{fg}}(x) = \mathrm{Re} \int_{-\infty}^{\infty} f(x + x')\, g^*(x')\, dx' \tag{6}$$

Substitute Equation (4) into Equation (6) to obtain

$$\int_{-\infty}^{\infty} C_{\mathrm{fg}}(x) e^{-iKx}\, dx = F(K) G^*(K) \tag{7}$$

Then numerical computation of the normalized form of $C_{\mathrm{fg}}(x)$ makes use of FFT (Fast Fourier Transform) algorithms via the sequence

$$C_{\mathrm{fg}}(x) = \mathcal{F}^{-1}\Big(\mathcal{F}(f(x))\, [\mathcal{F}(g(x))]^*\Big) \tag{8}$$

4. Semblance Method (SM)

Semblance is a generalization of cross-correlation and depends upon relative amplitudes in addition to correlation [25]. Assuming both signals f and g are real, the semblance $S_{\mathrm{fg}}(\tau)$ of two signals $f(t)$ and $g(t)$ is defined [25] as:

$$S_{\mathrm{fg}}(x) = \frac{\int_{-\infty}^{\infty} [f(x') + g(x' + x)]^2\, dx'}{2\left(\int_{-\infty}^{\infty} f^2(x') dx' + \int_{-\infty}^{\infty} g^2(x' + x)\, dx'\right)} \tag{9}$$

Define γ_f and γ_g as:

$$\begin{aligned} \gamma_f &= \int_{-\infty}^{\infty} f^2(x')\, dx', \\ \gamma_g &= \int_{-\infty}^{\infty} g^2(x')\, dx' \end{aligned} \tag{10}$$

Again it can be shown that $|S_{\mathrm{fg}}(x)| \leq 1$. From definitions in Equations (9) and (10) it follows that

$$S_{\mathrm{fg}}(x) = \frac{1}{2} + \frac{(\gamma_g/\gamma_f)^{1/2}}{1 + \gamma_g/\gamma_f} C_{\mathrm{fg}}(x) \tag{11}$$

Note $S_{\mathrm{fg}}(x)$ is linearly related to cross correlation $C_{\mathrm{fg}}(x)$ with gain coefficient of the form $\alpha/(1 + \alpha^2)$ having maximum value of $1/2$ for $\alpha = 1$, value 0 for $\alpha = 0$, and goes to zero as $1/\alpha$ for large α. This is an important property of SM depending on image signal-to-noise: only correlated signals of comparable amplitudes have large semblance.

5. Translation Phase Shift Method (TPSM)

The underlying idea of the translation phase shift method follows from definition in Equation (3). It is convenient to write this equation in relationship form, *i.e.*,

$$f(x) \Longleftrightarrow F(K) \tag{12}$$

From Equations (3) and (12) it follows that a translation of δ m in the space domain corresponds to a linear phase shift in the spatial frequency domain. In n-dimensions in $x_1, x_2, \cdots, x_n$ coordinates this relation is

$$f(\mathbf{x}+\delta) \Longleftrightarrow \exp(i\mathbf{K}\cdot\delta)\, F(\mathbf{K}) \tag{13}$$

For discrete FFT application with (N_x, N_y) point transforms in the x and y coordinates, Marchant *et al.* [23] defines the necessary FFT parameters.

Assume $f(x_n, y_m)$ and $g(x_n, y_m)$, $n = 1, 2, \cdots, N_x, m = 1, 2, \cdots, N_y$, are two successive digital N_x by N_y pixel images, where for vector field estimation, it is assumed that images f and g are related by translation. In the Fourier transform domain, the translational relationship of successive time-lapse images becomes

$$\mathrm{Im}(\log(G(K_x, K_y)/F(K_x, K_y))) = K_x\, \delta_x + K_y\, \delta_y \tag{14}$$

In Equation (14), Im denotes imaginary part. Let $G_{nm} = G(K_{xn}, K_{ym})$, and similarly for F_{nm}. Then define

$$D_{nm} = \mathrm{Im}(\log(G_{nm}/F_{nm})) \tag{15}$$

and quadratic form $L(\delta_x, \delta_y)$ as

$$L(\delta_x, \delta_y) = \sum_{n,m} w_{nm} |D_{nm} - K_{xn}\delta_x - K_{ym}\delta_y|^2 \tag{16}$$

In Equation (16), the weights w_{nm} are normalized such that

$$\sum_{n,m} w_{nm} = 1 \tag{17}$$

Minimization of Equation (16) leads to the linear weighted least-mean-square solution for the translational shifts δ_x, δ_y

$$\begin{pmatrix} \ell_{11} & \ell_{12} \\ \ell_{21} & \ell_{22} \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \tag{18}$$

where the elements are

$$\begin{aligned} \ell_{11} &= \textstyle\sum_{n,m} w_{nm} K_{xn}^2 \\ \ell_{12} &= \textstyle\sum_{n,m} w_{nm} K_{xn} K_{ym} \\ \ell_{21} &= \ell_{12} \\ \ell_{22} &= \textstyle\sum_{n,m} w_{nm} K_{ym}^2 \end{aligned} \tag{19}$$

and the right-hand-side elements are

$$\begin{aligned} b_1 &= -\textstyle\sum_{n,m} w_{nm} D_{nm} K_{xn} \\ b_2 &= -\textstyle\sum_{n,m} w_{nm} D_{nm} K_{ym} \end{aligned} \tag{20}$$

This method directly extends to three dimensions determining $\delta_x, \delta_y, \delta_z$ by introducing three-dimensional data matrices D_{nmp}.

6. Segmentation

CCM, SM and TPSM as formulated here produce global estimates of translation shifts δ_x, δ_y. Local estimates are obtained by segmenting the images into sub-domains. For FFT methods it is important for sub-domains to have FFT parameters N_x and N_y be at least 16 or 32 for reliable estimation. Because zero frequency is near the center of the transform sequence, and higher edge frequencies are typically noisy and not well resolved, only the low frequency central components are used in computing the matrix elements defined by Equations (19) and (20).

For a segmented implementation, let the digital image f_{nm}, $n = 1, 2, \cdots, n_r\,, m = 1, 2, \cdots, n_c$, be segmented into N_b^2 overlapping square sub-regions each having N_f rows and columns. Again for FFT application assume both N_b and N_f are compound integers. The distances between segment center row and column positions (N_{rs}, N_{cs})

$$\begin{aligned} N_{rs} &= \mathrm{floor((n_r - N_f)/(N_b - 1))} \\ N_{cs} &= \mathrm{floor((n_c - N_f)/(N_b - 1))} \end{aligned} \tag{21}$$

where floor(x) = greatest integer $\leq x$. Last distance between center positions given by Equation (21) needs to be adjusted if ratios in the arguments of the floor function are not integers. With the same proviso, left and right hand end points of pixel intervals of the subintervals $N_{rn}^{\mathrm{left}}, N_{rn}^{\mathrm{right}}, N_{cm}^{\mathrm{left}}, N_{cm}^{\mathrm{right}}$ are

$$\begin{aligned} N_{rn}^{\mathrm{left}} &= (n-1)N_{rs} + 1 \\ N_{rn}^{\mathrm{right}} &= (n-1)N_{rs} + N_f \\ N_{cm}^{\mathrm{left}} &= (m-1)N_{cs} + 1 \\ N_{cm}^{\mathrm{right}} &= (m-1)N_{cs} + N_f \end{aligned} \tag{22}$$

where $n, m, 1, 2, \cdots, N_b$. Slightly more complicated rules apply when $\mathrm{floor(x)} \neq \mathrm{x}$. As an example segmentation, assume the image pixel size is 512 × 512, let $N_b = N_f = 32$. The segmentation results in an output matrix of 32 × 32 with row and column distance between centers of $N_{rs} = N_{cs} = 15$.

7. Spatio-Temporal Method (STM)

A limitation of CCM, SM and TPSM is that intrinsic resolution can be less than the underlying time-lapse image sequence data. This is a consequence of transform methods requiring minimum sub-interval lengths of 16 or 32 to have reliable central transform values. STM is formulated in the space and time domain. Because of this, STM honors the resolution intrinsic to the data. The method, also called optical flow, is well documented by Lim [26]. The method is actually related to a more general conservation law in phase space, namely Liouville's theorem from statistical mechanics [27].

In common with the other three methods, time-lapse image differences are assumed to be caused solely by aerosol feature pattern translation. For three-dimensional motion, with local velocity components (v_x, v_y, v_z), this assumption leads to the relationship between consecutive image frames at times t_{n-1} and $t_n = t_{n-1} + \Delta t$ for $t_{n-1} \leq t \leq t_n$. Equation (23) is valid for small enough time increments $\Delta t = t_n - t_{n-1}$ and over source-free regions where aerosol patterns are advected without distortion. As shown in [26], Equation (23) satisfies the first order partial differential equation

$$\begin{aligned} f(x,y,z,t) &= f(x-v_x(t-t_{n-1}), y-v_y(t-t_{n-1}) \\ & \quad z-v_z(t-t_{n-1}), t_{n-1}) \end{aligned} \tag{23}$$

$$\begin{aligned} \tfrac{\partial f(x,y,z,t)}{\partial x} v_x + \tfrac{\partial f(x,y,z,t)}{\partial y} v_y & \\ + \tfrac{\partial f(x,y,z,t)}{\partial z} v_z + \tfrac{\partial f(x,y,z,t)}{\partial t} &= 0 \end{aligned} \tag{24}$$

Equation (24) can be used to derive a system of equations for the local velocity component estimates (v_x, v_y, v_z). Use segmentation as developed in Section 6, with N_f a small odd integer, for example 5 or 7 for voxel spatial coordinates (x_m, y_n, z_p), and two successive time values $t_{\ell-1}, t_\ell$. Then define the quadratic cost function $\mathcal{L}(v_x, v_y, v_z)$ as

$$\begin{aligned} \mathcal{L}(v_x, v_y, v_z) = \textstyle\sum_{m,n,p,\ell} | \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial x} v_x + \\ \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial y} v_y + \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial z} v_z + \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial t} |^2 \end{aligned} \tag{25}$$

The multi-dimensional sums in Equation (25) are over local neighborhoods. Minimization of $\mathcal{L}(v_x, v_y, v_z)$ with respect to the local velocity components yields the matrix equation

$$\begin{pmatrix} \mathrm{L}_{11} & \mathrm{L}_{12} & \mathrm{L}_{13} \\ \mathrm{L}_{21} & \mathrm{L}_{22} & \mathrm{L}_{23} \\ \mathrm{L}_{31} & \mathrm{L}_{32} & \mathrm{L}_{33} \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{26}$$

where the symmetric matrix elements are defined as

$$\begin{aligned} \mathrm{L}_{11} &= \textstyle\sum_{m,n,p,\ell} (\tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial x})^2 \\ \mathrm{L}_{12} &= \textstyle\sum_{m,n,p,\ell} \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial x} \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial y} \\ \mathrm{L}_{13} &= \textstyle\sum_{m,n,p,\ell} \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial x} \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial z} \\ \mathrm{L}_{21} &= \mathrm{L}_{12}\,, \\ \mathrm{L}_{22} &= \textstyle\sum_{m,n,p,\ell} (\tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial y})^2 \\ \mathrm{L}_{23} &= \textstyle\sum_{m,n,p,\ell} \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial y} \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial z} \\ \mathrm{L}_{31} &= \mathrm{L}_{13} \\ \mathrm{L}_{32} &= \mathrm{L}_{23} \\ \mathrm{L}_{33} &= \textstyle\sum_{n,m,p,\ell} (\tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial z})^2 \end{aligned} \tag{27}$$

Similarly, the right-hand-side elements of Equation (26) are

$$\begin{aligned} b_1 &= -\textstyle\sum_{m,n,p,\ell} \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial x} \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial t} \\ b_2 &= -\textstyle\sum_{m,n,p,\ell} \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial y} \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial t} \\ b_3 &= -\textstyle\sum_{m,n,p,\ell} \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial z} \tfrac{\partial f(x_m,y_n,z_p,t_\ell)}{\partial t} \end{aligned} \tag{28}$$

Equation (26) is solved for each space-time neighborhood center. Solution in Equation (26) is implemented in the space-time domain allowing small neighborhoods to be employed yielding resolution depending only upon the data. Because Equation (26) may be poorly conditioned, truncated singular value decomposition (TSVD) [28] can be used as implemented in the computational and graphical environment Matlab [29]. In examples computed for this report, the condition number C_n is not an issue. The order of magnitude of C_n is ≈ 10.

In the three-dimensional case, an equation of motion, namely conservation of mass or mass balance [30], can be applied to stabilize the solution. Let $\rho(x, y, z, t)$ be the atmospheric aerosol number density in number of particles per unit volume, then the conservation law for the velocity field $\mathbf{v}(x, y, z, t)$ of the tracer particles is written in the familiar form

$$\nabla \cdot (\rho(x, y, z, t)\mathbf{v}(x, y, z, t)) + \frac{\partial \rho(x, y, z, t)}{\partial t} = 0 \tag{29}$$

This general result often simplifies to the condition of incompressible flow. This approximation of microscale meteorological conditions is valid [30] when distances $L << 12$ km, wind speeds v $<< 100\text{m/s}$, L$<<$ v_s^2/g, and L$<<$ v_s/f, where the air velocity of sound $v_s \approx 331.4 + 0.6\,T_c$ [m/s], $g = 9.81\text{m/s}^2$ is the typical gravitational constant acceleration at the earth's surface, T_c is the temperature in °C, and f is the frequency in Hz of possible pressure waves. These conditions are almost always met for microscale wind fields. When these conditions are true, incompressible flow results and it follows that

$$\nabla \cdot \mathbf{v}(x, y, z, t) = 0 \tag{30}$$

Implementation of Equation (30) in STM couples neighborhood solutions as discussed in [23].

8. Image Filtering and Processing

In order to improve image resolution in the time-lapse data, some image processing and filtering can be applied. For time lapse image analysis for motion retrieval, a good indication of image resolution is the relative area under local semblance peaks in the vicinity of local maxima. An open question is the importance of quantization error in this regard, *i.e.*, do 12 bit gray scale images with 4,096 levels significantly outperform 8 bit gray scale images. Related to this is the importance of image histogram equalization Schols and Eloranta [17] and to better utilize the image spectrum Piironen and Eloranta [12].

In Schols and Eloranta [17] high pass temporal filtering is used to remove structures that do not move with the wind. For wind farm citing applications, ground returns from towers, buildings, telephone poles, *etc.*, are removed by mapping the LIDAR field of view at the beginning of the survey. Here, too, the short term temporal variability due to turbulence is important for estimating wind turbine component fatigue.

The first three methods process in the spatial frequency domain. In these cases, optional FFT based filters are used. Kaiser low pass windows [26] with adjustable side lobe level and bandwidth are employed. In the STM, filtering and signal processing are implemented directly in the space domain. For reasons of efficiency, the space domain filters use an auxiliary large redundant matrix. If input image matrix f_{in} has size $[n_r, n_c]$, then the derived auxiliary matrix $f_{\text{aux}}(i, j)$ has size $[n_r n_c, n_{\text{neib}}]$ where n_{neib} is the number of neighborhood pixels for the image point i, j. This approach trades computer memory for speed of execution. All segmented filter and image processing are then efficient dot products. With this approach, in Matlab syntax, application of filter $\mathcal{F}$ to input matrix f_{in} then is simply the dot product

$$\mathcal{F} f_{\text{in}} = \text{reshape}(f_{\text{aux}} * w', n_r, n_c) \tag{31}$$

where w is the one-dimensional row vector form (of length n_{neib}) of the filter coefficients for one, two, or three-dimensional filtering and reshape returns the one dimensional output into the original matrix size with dimensions $[n_r, n_c]$. Near edges of images, the auxiliary matrix $f_{\mathrm{aux}}(i, j)$ is augmented with duplicated values extending outside of the image domain in order to define all necessary nearest neighbor pixels. Note that as a consequence, the corner regions of image domains have poorly conditioned neighborhood matrices. In the three-dimensional case, neighborhood extension off the bounding surfaces is easily and efficiently implemented with that Matlab repmat function. Note that the spatio-temporal neighborhood averaging explicit in matrix element definitions in Equation (27), the large augmented matrix f_{aux} as defined by Equation (31), is not needed. Matlab vector subscripts are used to index local neighborhoods of the input image f_{in}.

9. Dynamical Aerosol Cloud Model

To develop robust wind field estimation methods, it is essential to have a realistic dynamical moving aerosol density model. The model should be stochastic, turbulent, and produce a known underlying wind field allowing space-time error analysis.

Fortunately, such models exist. In particular, we make use of the pioneering and seminal work of Stam [19,20]. His method uniquely results in succinct and efficient algorithms. We implement a version of his stochastic vector field model using a Kolmogorov wave number energy spectrum [31] specifying the distribution of energy and size scale of turbulent vortices.

The turbulent model of Stam incorporated here is an efficient FFT-PC based solution to the nonlinear Navier–Stokes equation for incompressible fluids with periodic boundary conditions. Reference to some of his numerical result graphics confirms their realism. Simulating fluids is one of the more challenging problems in computational physics so this is a non-trivial feature. Even so, the model does not include temperature effects, vertical wind shear caused by surface friction arising from vegetation, terrain, *etc.* However we believe it is adequate for evaluating different algorithms for dynamic wind field retrievals. The physics of turbulent flow between the earth's surface and the boundary layer is more complicated, see Kaimal and Finnigan [32], Kaimal, Wyngaard, Izumi, and Cote [33]. If specific terrain and roughness effects are required, they can be indirectly estimated using the commercial linearized software WAsP (Wind Atlas Analysis and Application Program), developed by DTU (Technical University of Denmark) and the Risø National Laboratory. A future non-linear release of the WAsP will incorporate a full CFD treatment of turbulence.

The wind field is composed of two components, large and small, *i.e.*,

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_\ell(\mathbf{x}, t) + \mathbf{u}_s(\mathbf{x}, t) \tag{32}$$

The large scale is chosen to be slowly varying and deterministic. The small scale is random and turbulent. Dropping its subscript, this vector field component has the Fourier transform representation

$$\mathbf{u}(\mathbf{x}, t) = \int \frac{d^3K}{(2\pi)^3} \int \frac{d\omega}{2\pi} e^{i(\mathbf{K}\cdot\mathbf{x}-\omega t)}\, \mathbf{U}(\mathbf{K}, \omega) \tag{33}$$

Under the Gaussian assumption, the random component is defined by its first two statistical moments. Without loss of generality, the means are assumed to be zero. The remaining statistic is the space-time cross correlation $\phi_{ij}(\mathbf{x}, t)$ between wind components i and j defined as

$$\phi_{ij}(\mathbf{x}, t) = \int d^3x' \int dt' < u_i^*(\mathbf{x} - \mathbf{x}', t - t') u_j(\mathbf{x}', t') > \tag{34}$$

By the correlation theorem the Fourier transform of $\phi_{ij}(\mathbf{x}, t)$ is

$$\Phi_{ij}(\mathbf{K}, \omega) =< U_i^*(\mathbf{K}, \omega) U_j(\mathbf{K}, \omega) > \tag{35}$$

The wind field is assumed to be incompressible, *i.e.*,

$$\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0 \tag{36}$$

There is a simple way to enforce the incompressibility condition in the Fourier transform domain. By the Helmholtz theorem [34], any suitable continuous vector field $\mathbf{v}(\mathbf{x}, t)$ can be decomposed into transverse and longitudinal components. The Helmholtz decomposition [35] yields the longitudinal component

$$\mathbf{v}_\ell(\mathbf{x}, t) = -\frac{\nabla\nabla}{4\pi} \cdot \int d^3x' \frac{\mathbf{v}(\mathbf{x}', t)}{|\mathbf{x} - \mathbf{x}'|} \tag{37}$$

In the Fourier domain this becomes

$$\mathbf{V}_\ell(\mathbf{K}, \omega) = \frac{\mathbf{K}\mathbf{K} \cdot \mathbf{V}(\mathbf{K}, \omega)}{k^2} \tag{38}$$

From this it follows that the projection operator P_{ij} onto the transverse component of the vector field is

$$P_{ij} = \frac{(k^2\delta_{ij} - K_iK_j)}{k^2} \tag{39}$$

where δ_{ij} is the Kronecker delta

$$\delta_{ij} = \begin{cases} 1\,, \mathrm{if\ i = j} \\ 0\,, \mathrm{otherwise} \end{cases} \tag{40}$$

Thus for any vector field $\mathbf{v}(\mathbf{x}, t)$, the projected vector field $\mathbf{u}(\mathbf{x}, t)$

$$\mathbf{u}(\mathbf{x}, t) = \mathcal{F}^{-1} \sum_{i,j=1}^{3} \mathbf{i}_i P_{ij} V_j(\mathbf{K}, \omega) \tag{41}$$

is transverse, *i.e.*, $\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0$. In Equation (41) $\mathbf{i}_i$ is the unit vector in the direction x_i, and $\mathcal{F}^{-1}$ is the inverse Fourier transform operator. The mean kinetic energy per unit mass of the vector field is

$$\mathcal{E} = 1/2 \int d^3x \int dt\ < \mathbf{u}^*(\mathbf{x}, t) \cdot \mathbf{u}(\mathbf{x}, t) > \tag{42}$$

with the alternative frequency domain form

$$\begin{aligned} \mathcal{E} &= 1/2 \int \tfrac{d^3K}{(2\pi)^3} \int \tfrac{d\omega}{2\pi} < \mathbf{U}^*(\mathbf{K}, \omega) \cdot \mathbf{U}(\mathbf{K}, \omega) > \\ &= \int \tfrac{d^3K}{(2\pi)^3} \int \tfrac{d\omega}{2\pi} A(\mathbf{K}, \omega) \\ &= \int_0^\infty dk \int \tfrac{d\omega}{2\pi} E(k, \omega) \end{aligned} \tag{43}$$

The kinetic energy spectrum function $E(k, \omega)$ as defined in Equation (43) results from the isotropic assumption and is thus defined in terms of the energy amplitude $A(\mathbf{K}, \omega)$ as

$$E(k, \omega) = \frac{A(\mathbf{K}, \omega)}{4\pi k^2} \tag{44}$$

In the case of isotropic, locally homogeneous turbulence, Lesieur [31] determines the relationship between the kinetic energy spectrum function and the cross-spectral density function, namely

$$\Phi_{ij}(\mathbf{K},\omega) = \begin{cases} P_{ij}\,\frac{E(k,\omega)}{4\pi k^2}\ , \text{in 3 dimensions} \\ P_{ij}\,\frac{E(k,\omega)}{\pi k}\ , \text{in 2 dimensions} \end{cases} \tag{45}$$

where $k^2 = K_1^2 + K_2^2 + K_3^2$. Equation (45) assumes also for the three dimensional case the wind field helicity is zero. The famous Kolmogorov turbulent flow energy spectrum can be used to define $E(k,\omega)$ when the Reynolds number is asymptotically large. Measurements of ocean and atmospheric turbulence show that this spectrum fits remarkably well over several orders of magnitude of spatial wavenumber. As shown for example by [36], the form of the spectrum can be determined by a dimensional argument when $E(k,\omega)$ depends only on the energy viscous dissipation rate ε and the wavenumber magnitude k. This yields the Kolmogorov spatial wavenumber energy spectrum

$$E_K(k) = \begin{cases} 0, & \text{, for } k < k_i \\ C_K\varepsilon^{2/3}/k^{5/3} & \text{, otherwise} \end{cases} \tag{46}$$

As stated by [37], *the energy spectrum of fully developed homogeneous turbulence is thought to be composed of three regions: a region of energy injection at the largest scales, an intermediate inertial range characterized by zero forcing and zero dissipation, and, at the very smallest scales, a region dominated by viscosity.* The interpretation of Equation (46) is that energy is injected by large eddies with wave numbers near the inertial value k_i. The energy cascades to smaller and smaller eddies where it is removed by molecular viscosity.

Using Stam's method [19], the kinetic energy spectrum function $E(k,\omega)$ is defined as

$$E(k,\omega) = E_K(k)\,G_k(\omega) \tag{47}$$

where $G_k(\omega)$ is the Gaussian

$$G_k(\omega) = \frac{1}{(2\pi)^{1/2}k\sigma}\ \exp\Big(\frac{-\omega^2}{2k^2\sigma^2}\Big) \tag{48}$$

Note that for this choice

$$\int_{-\infty}^{\infty} E(k,\omega)d\omega = E_K(k) \tag{49}$$

showing that kinetic energy is conserved. The form of $G_k(\omega)$ also causes small eddies (large k) to correctly damp out more quickly than large ones.

The second order statistics for the wind field are now determined. It remains to construct an explicit associated wind field $\mathbf{u}(\mathbf{x},t)$ and a dynamic aerosol mass density $\rho(\mathbf{x},t)$. Again following [19], a direct route to this end assumes the velocity spectrum function defined by Equation (33) is given via deterministic kernel functions $\mathrm{H}_{\mathrm{n}\ell}(\mathbf{K},\omega)$, *i.e.*,

$$\mathbf{U}_n(\mathbf{K},\omega) = \sum_{\ell=1}^{3} \mathrm{H}_{\mathrm{n}\ell}(\mathbf{K},\omega)\, w_\ell(\mathbf{K},\omega) \tag{50}$$

where $w_\ell(\mathbf{K},\omega)$ are white noise spectral functions with statistical expectations

$$< w_\ell(\mathbf{K},\omega)\, w_{\ell'}(\mathbf{K},\omega) >= \delta_{\ell\,\ell'} \tag{51}$$

Thus substitution of Equation (50) into Equation (35) yields

$$\Phi_{nm}(\mathbf{K},\omega) = \sum_{\ell=1}^{3} \mathrm{H}_{\mathrm{n}\ell}(\mathbf{K},\omega)\, \mathrm{H}_{\mathrm{m}\ell}(\mathbf{K},\omega) \tag{52}$$

Because $\Phi_{nm}(\mathbf{K},\omega)$ is symmetric, only six elements in $\mathrm{H}_{\mathrm{n}\ell}$ are independent. Let $\mathrm{H}_{12} = \mathrm{H}_{13} = \mathrm{H}_{23}$ = 0 resulting in the solution

$$\begin{array}{rcl} \mathrm{H}_{11} &=& \sqrt{\Phi_{11}} \\ \mathrm{H}_{21} &=& \Phi_{21}/\mathrm{H}_{11} \\ \mathrm{H}_{22} &=& \sqrt{\Phi_{22} - \mathrm{H}_{21}^2} \\ \mathrm{H}_{31} &=& \Phi_{31}/\mathrm{H}_{11} \\ \mathrm{H}_{32} &=& (\Phi_{32} - \mathrm{H}_{31}\,\mathrm{H}_{21})/\mathrm{H}_{22} \\ \mathrm{H}_{33} &=& \sqrt{\Phi_{33} - \mathrm{H}_{31}^2 - \mathrm{H}_{32}^2} \end{array} \tag{53}$$

Using a numerical pseudo-random number generator define the random vectors

$$\mathrm{X}_\ell = \mathrm{r}_\ell\, \mathrm{e}^{2\mathrm{i}\theta_\ell} \tag{54}$$

where r_ℓ is a normally distributed and θ_ℓ is uniformly distributed on the unit interval $[0, 1]$. Both are four-dimensional arrays of size (N_x, N_y, N_z, N_t). This results in the explicit velocity spectra

$$U_n(\mathbf{K},\omega) = \sum_{\ell=1}^{3} \mathrm{H}_{\mathrm{n}\ell}(\mathbf{K},\omega)\, \mathrm{X}_\ell(\mathbf{K},\omega) \tag{55}$$

and finally, by inverse Fourier transform, the desired model for the turbulent four-dimensional space-time velocity field

$$u_n(\mathbf{x},t) = \mathcal{F}^{-1} U_n(\mathbf{K},\omega) \tag{56}$$

10. Aerosol Transport

It remains to associate with the velocity field $\mathbf{u}(\mathbf{x},t)$ an approximate dynamic aerosol mass density function $\rho(\mathbf{x},t)$. A complete atmospheric aerosol model accounts for advection, diffusion, nucleation, coagulation, condensation growth, evaporation, sedimentation and aqueous chemistry [38,39]. These references point to the complicated and as yet not fully understood atmospheric dynamics of aerosol chemistry and physics. However, for short periods of time (a few seconds), many of the processes can be neglected. The fundamental mass conservation law is

$$\frac{\partial\rho(\mathbf{x},t)}{\partial t} + \nabla\cdot\mathbf{j}(\mathbf{x},t) = S(\mathbf{x},t) \tag{57}$$

In Equation (57) $S(\mathbf{x},t)$ represents aerosol sources and sinks and $\mathbf{j}(\mathbf{x},t)$ is the mass current. For aerosol transport there are two types of current, one advective and the other diffusive [40]. Thus

$$\begin{aligned} \mathbf{j}(\mathbf{x},t) &= \mathbf{j}_{\text{adv}}(\mathbf{x},t) + \mathbf{j}_{\text{diff}}(\mathbf{x},t) \\ \mathbf{j}_{\text{adv}}(\mathbf{x},t) &= \mathbf{u}(\mathbf{x},t)\,\rho(\mathbf{x},t) \\ \mathbf{j}_{\text{diff}}(\mathbf{x},t) &= D\,\nabla\rho(\mathbf{x},t) \end{aligned} \tag{58}$$

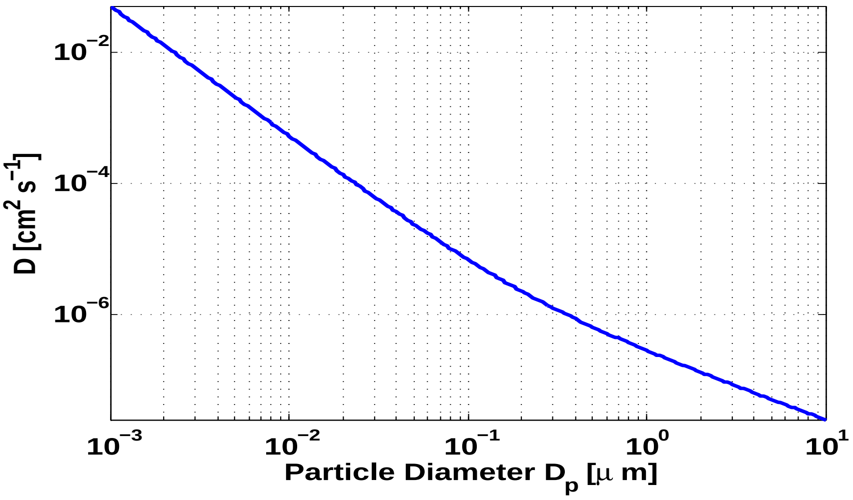
In Equation (58), D is the diffusion coefficient for aerosol particles in the Stokes regime. For incompressible flow the resulting *advection-diffusion equation* is

$$\frac{\partial\rho(\mathbf{x},t)}{\partial t} + \mathbf{u}(\mathbf{x},t)\cdot\nabla\rho(\mathbf{x},t) - \nabla\cdot(D\nabla\rho(\mathbf{x},t)) = S(\mathbf{x},t) \tag{59}$$

The diffusion coefficient D is dependent on the particle diameter D_p ([5]; p. 474). Figure 1 shows the dependence of D as a function of the aerosol particle diameter D_p in μm. To estimate the relative importance of the advective and diffusion terms, assume that the wind field is a constant $\mathbf{u} = \mathbf{u_0}$. Taking three dimensional Fourier transforms of the two terms over the spatial dependence then yields

$$\begin{aligned} \mathcal{F}\Big(\mathbf{u}(\mathbf{x},t)\cdot\nabla\rho(\mathbf{x},t) - \nabla\cdot(D\nabla\rho(\mathbf{x},t))\Big) &= (i\mathbf{u_0} + D\,\mathbf{K})\cdot\mathbf{K}\,P(\mathbf{K},t) \\ \mathcal{F}\Big(\rho(\mathbf{x},t)\Big) &= P(\mathbf{K},t) \end{aligned} \tag{60}$$

Figure 1. Diffusion coefficient D as a function of aerosol diameter D_p in μm at 20 °C.



The spatial wavenumber magnitude is $K = |\mathbf{K}| = (K_x^2 + K_y^2 + K_z^2)^{1/2}$ $[\mathrm{m}^{-1}]$. Use a spatial grid of $\mathrm{d}x = \mathrm{d}y = \mathrm{d}z = 1$ m, and the digital transform relation $\mathrm{d}x\,\mathrm{d}K_x = 2\pi/N_x$, where N_x is the number of grids in the x direction and similarly for y and z directions. It follows that K has the maximum value $K_{\max} = 2\pi\,3^{1/2} = 10.9$. From Figure 1, the maximum diffusion coefficient $D_{\max}$ for particles with $D_p = 1 \times 10^{-3}\mu$m is $D_{\max} \approx 2 \times 10^{-6}\,\mathrm{m}^2/\mathrm{s}$ so that $K_{\max}\,D_{\max} \approx 2.2 \times 10^{-5}\mathrm{m/s}$. The maximum Brownian motion transport speed is four orders of magnitude less than usable surface winds. Hence the diffusion term in Equation (59) can be neglected. In the absence of diffusion and sources or sinks, the advective solution to Equation (59), from Equations (23) and (24) is

$$\rho(\mathbf{x},t+dt) = \rho(\mathbf{x} - \mathbf{u}\,dt, t) \tag{61}$$

Statistically the total mass density of aerosol particles is a measured quantity, *i.e.*, the total suspended particulates or TSP. Reference Brook *et al.* [41] is based upon the Canadian National Air Pollution Surveillance network, and at the time was one of the world's largest and most geographically diverse data sets. The average 24 h collections at 14 sites with a sample size of n = 2,831 resulted in a TSP normal distribution $N(m, \sigma)$ fit with mean $m = 55\,\mu\,\mathrm{g/m}^3$ and standard deviation $\sigma = 38\,\mu\,\mathrm{g/m}^3$. Our wind field uses $m = 100\,\mu\,\mathrm{g/m}^3$ and $\sigma = 20\,\mu\,\mathrm{g/m}^3$.

Let $\rho_n(\mathbf{x}) \equiv \rho(\mathbf{x}, t_n)$ be the mass density at the n^{th} time interval and $\Delta\rho(\mathbf{x})$ a small random source of aerosol particulates. Then the density model is defined as

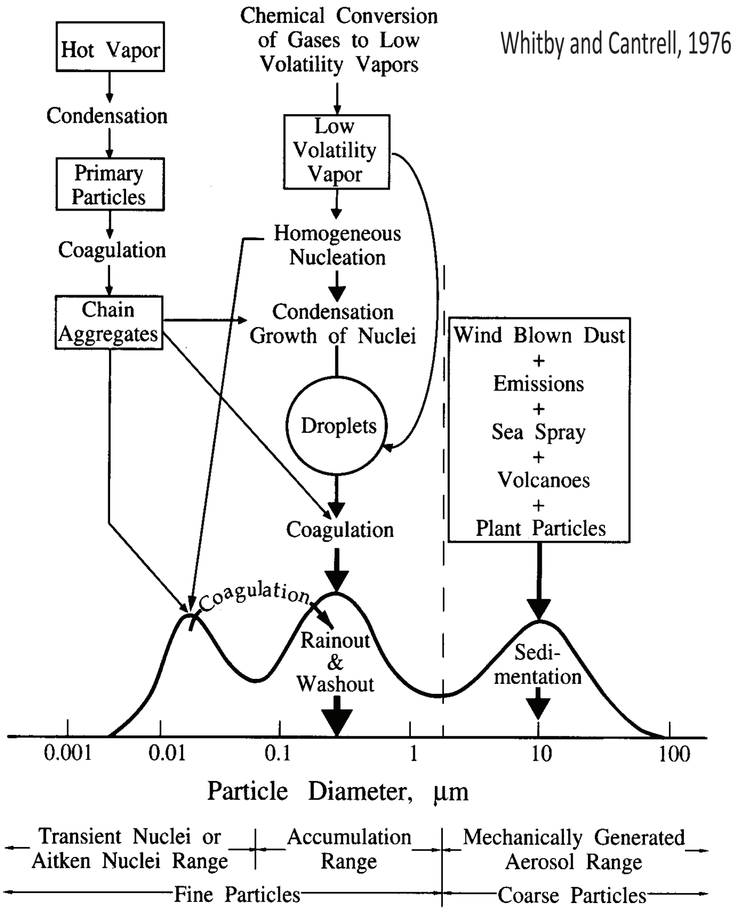
$$\rho_n(\mathbf{x}) = \rho_{n-1}(\mathbf{x} - \mathbf{u}_{n-1}(\mathbf{x})\Delta t) + \Delta\rho(\mathbf{x}) \tag{62}$$

The refresh term in Equation (62) is necessary. Without it, after a few cycles, the advection process mixes the atmospheric aerosol density distribution to a non-physical featureless uniform one. Particles in the atmosphere are continuously being created and removed by several complicated chemical processes including nucleation, coagulation, and Stokes sedimentation [5]. More importantly, wind blown sand and minerals lofted by saltation is a primary source of atmospheric mineral dust [42]. Recently it has been discovered that the saltated particles are ionized, thus increasing this effective particle source [43]. Figure 2, Whitby and Cantrell [44] gives an idealized tri-modal distribution of atmospheric aerosol surface area as a function of particle diameter in $\mu\,$m. It shows how the three particle size modes interact, and contains particle sources and sinks. The particles in the accumulation range, from 0.1 to 2.5 $\mu\,$m, have the longest lifetimes and also represent the largest component of aerosol surface area per unit volume of air. The coarse range particles, created from wind blown dust and biological particles as well as anthropomorphic sources, have relatively brief lifetimes because of their larger Stokes settling time. For example particles of diameter greater than 10 μm settle with speeds greater than 10 $\mathrm{m/h}$. However this mode of the particle size distribution has much larger volume backscatter coefficients for $1.5\,\mu\,m$ laser light.

In Equation (62), the initial time value $\rho_1(\mathbf{x})$ is chosen to be a three-dimensional (N_x, N_y, N_z) random normally distributed array with given mean and standard deviation (m_m, σ_m), *i.e.*, distributed $N(m_m, \sigma_m)$ representing TSP (total suspended particulates) in units of $\mu g/m^3$. As seen in Figure 2, atmospheric aerosol TSP typically contains particle diameters ranging from 0.01 to 100 μm. Mean value mass densities are in the range $20 < m_m < 200\,\mu g/m^3$. At later times, $n = 2, 3, \cdots, N_t$, the density is advected using interpolation. Points advected outside the rectangular volume are wrapped around by periodic extension of the fundamental cell. Without the random addition of $\Delta\rho(\mathbf{x})$ to Equation (62), numerical diffusion would flatten the starting distribution to homogeneity. The heterogeneity of the aerosol mass density function $\rho_n(\mathbf{x})$ makes it possible to correlate aerosol density fluctuations.

Because of the cited complicated nature of actual aerosol dust sources and their interactions, a simple ad hoc source model is chosen. To maintain aerosol heterogeneity, a small amplitude of random mass density $\Delta\rho(\mathbf{x})$, 3.5 percent of the mean value of TSP, is injected with random subsets of pixels into the fundamental cell at each time frame. This amplitude value is chosen empirically to maintain the spatial bandwidth of the aerosol density images approximately constant. The injected TSP also draws from distributions N(m,s). One half the points receive this injection. The grid injection points are also selected randomly for each cycle. For each cycle, the modified distribution is renormalized to N(m,s).

Figure 2. Idealized schematic of atmospheric aerosol cycles including sources, sinks, particle modes and particle creation as a function of particle diameter in μm. From Whitby and Cantrell [44].



11. Streamlines

Recently, the challenge to visualize high-resolution dynamic three-dimensional vector fields in an intuitive manner has been addressed by many authors (see for example [21,22,45]). For static vector field visualization, high resolution spatial coherence of the vector field is displayed by using a random texture background with essentially zero correlation length. Line integral convolution (LIC) along the streamlines (field lines) correlate the image along field lines, literally like the elementary experiment using iron fillings on a piece of glass to visualize the magnetic field lines of a bar magnet placed beneath the glass. This results in spatial coherence along flow lines. For high-resolution dynamic vector field visualization, the random background texture map evolves continuously in time to support temporal coherence [21]. Here we choose a more simple approach and use the aerosol movie density maps as background, and superimpose computed streamlines over the density maps.

A streamline or flow line is a path $\boldsymbol{\sigma}(\boldsymbol{x}, \boldsymbol{x}_0)$, with start point x_0. By definition a streamline is everywhere tangent to the vector field $\mathrm{v}(\mathrm{x})$. Physically, a streamline is the path that a massless particle would follow. Let $x(u), y(u), z(u)$ be a parametric representation of a streamline. Streamlines are thus solutions to the first order vector differential equation

$$\frac{d\boldsymbol{\sigma}(u)}{du} = \mathbf{v}(\boldsymbol{\sigma}(u)) \tag{63}$$

Solutions $\boldsymbol{\sigma}(u)$ to Equation (63) beginning at point $\mathbf{x}_0$ are called integral curves. It can be shown that in regions where $|\mathbf{v}| > 0$, the solutions are unique [46]. Equation (63) is simplified when the differential parameter du is chosen as the arc length differential $ds = (dx^2 + dy^2 + dz^2)^{1/2}$. Units are chosen such that the speed is $v = (x_u^2 + y_u^2 + z_u^2)^{1/2}$ resulting in

$$\frac{d\boldsymbol{\sigma}(u)}{ds} = \hat{\mathbf{v}}(\boldsymbol{\sigma}(u)) \tag{64}$$

where $\hat{\mathbf{v}}$ is the unit vector pointing in direction $\mathbf{v}$. The start points $\mathbf{x}_0$ are chosen to be a randomly permuted subset N_0 of the centers of the display grid cells, and then each point is moved to a random location within its cell. Numerical solutions to streamline Equation (64) typically use an error correcting Runge–Kutta algorithm [47]. Experimentation shows that a Runge–Kutta 23 solver, which compares point wise second and third order solutions for error correction, is a reasonable compromise between speed and accuracy. For our application, the right-hand-side unit vector in Equation (64) is determined by interpolation, in the examples to follow by two-dimensional interpolation of the velocity field estimated on a three-dimensional grid by the spatio-temporal method. Profiling shows interpolation is the most time consuming task in streamline computation. In our Matlab [29] script, substitution, where possible, of $\mathrm{interp2}$ by $\mathrm{qinterp2}$ [48] reduces computation time by approximately a factor of six.

12. Summary of Orbiting Cloud Model

Reference [23] compares the four wind field estimation methods using a two-dimensional cloud model. The particle density of the cloud is Gaussian modulated, with a filtered random-like shape. The cloud background is filtered random noise. The cloud is stretched and rotated to have a prescribed aspect ratio. The centroid of the cloud moves in a circular orbit. Forty five time lapse 2D images (512 × 512 pixels) corresponding to one orbit are used to simulate a moving cloud. Eight figures compare the true centroid velocity components (v_x, v_y) in units of m/s as a function of time with the estimated values $(\hat{v}_x, \hat{v}_y)$ for each of the four methods. All comparisons are based upon the same 45 image data set. Because of the circular motion, this results in a noisy sine wave versus time for v_x and similarly a noisy cosine wave for v_y. For this model, the true velocity is assumed to be defined by the cloud centroid motion. The cloud edge is modulated by a small amplitude random number input that varies image to image. Thus all four methods, based upon all pixel information in the images, will be affected. Four conclusions can be drawn from this study. Firstly, SM is superior to CCM, because of the ability of SM to discriminate against noise. Secondly, for STM, 2D local neighborhood median filtering yields better estimates than the equivalent 2D local neighborhood mean filtering. Thirdly, on average, SM and STM provide better estimates than CCM and TPSM. The fourth observation is that for TPSM, for this type of noisy cloud model, a global 2D Kaiser window, *i.e.*, using segmentation parameters $(N_f = 512,\ N_b = 1)$, yields more accurate estimates than the choice $(N_f = 384 = 128 \times 3,\ N_b = 6)$. This is because the underlying centroid motion is composed of low spatial frequencies.

Table 1 statistically summarizes the time statistics of median, mean, and standard deviation for STM velocity estimation over the the 45 time lapse images for the velocity field parameters v_x, v_y and speed

$|\mathbf{v}| = (v_x^2 + v_y^2)^{1/2}$. Initially each (512 × 512) image is median filtered using a moving (9 × 9) mask. Thus the pixel value in the center of the mask is replaced by the median value for the (9 × 9) neighborhood. The segmentation parameters are $(N_f = 16,\ N_b = 64)$. Image signal-to-noise ratio is 5 dB.

For this type of motion detection, the wind field estimation uses threshold speeds greater than 2 m/s. To discriminate against noise or and low level eddies, define precomputed local neighborhood sums $q(n_x, n_y, n_z, \ell)$ centered on voxel (n_x, n_y, n_z) of the form

$$q(n_x, n_y, n_z, \ell) = \sum_{m,n,p,\ell} \left(\frac{\partial f(x_m, y_n, z_p, t_\ell)}{\partial t} f(x_m, y_n, z_p, t_\ell)\right)^2 \tag{65}$$

For time intervals defined by subscript ℓ, $q(n_x, n_y, n_z, \ell)$ is sorted in descending order over each image segment. Larger values correspond to good signal associated with significant motion. Summary statistics are computed to compare STM and SM processing. For comparisons, the random generator seed for computing the random component of the cloud centroid position was fixed to provide the same model values for the true values of v_x, v_y and v. In Table 1, estimates are denoted $\hat{v}_{grd}$. For centroid motion, the velocities of the five largest segmentation neighborhood values of $|\mathbf{v}|$ are averaged to define centroid motion estimates in Table 1 referred to as $\hat{v}_{max}$. The true values for median, mean, and standard deviation statistics are under the columns labeled $\mathrm{med}(\mathrm{v})$, $\mathrm{avg}(\mathrm{v})$ and $\mathrm{std}(\mathrm{v})$.

Table 1. Summary statistics for STM processing averaged over all 45 orbiting cloud images.

	$\mathrm{med}(\mathrm{v})$	$\mathrm{med}(\hat{\mathrm{v}}_{\mathrm{max}})$	$\mathrm{med}(\hat{\mathrm{v}}_{\mathrm{grd}})$	$\mathrm{avg}(\mathrm{v})$	$\mathrm{avg}(\hat{\mathrm{v}}_{\mathrm{max}})$	$\mathrm{avg}(\hat{\mathrm{v}}_{\mathrm{grd}})$	$\mathrm{std}(\mathrm{v})$	$\mathrm{std}(\hat{\mathrm{v}}_{\mathrm{max}})$	$\mathrm{std}(\hat{\mathrm{v}}_{\mathrm{grd}})$
v_x	–0.5551	–0.3789	–0.2495	–0.0059	0.1196	0.0716	10.1753	9.3823	8.8252
v_y	–1.0219	–0.9864	–1.0063	–0.2116	–0.2216	–0.2247	6.0165	6.0044	6.0499
$\lvert\mathbf{v}\rvert$	11.8414	11.4639	11.0144	11.4624	10.8603	10.4340	2.3109	1.8587	1.7724

Table 1 shows that for the STM method, average speed estimates and standard deviation estimates for this case are accurate to within a few tenths of m/s with the v_{max} estimates doing slightly better. Table 2, using the same notation and cloud model shows the analogous results for SM processing. For this type of motion, SM processing uses global estimation, *i.e.*, centroid motion, without segmentation. For the SM case, the gradient processing is seen to be slightly better. Statistically the two methods for this model are seen to yield equivalent accuracy.

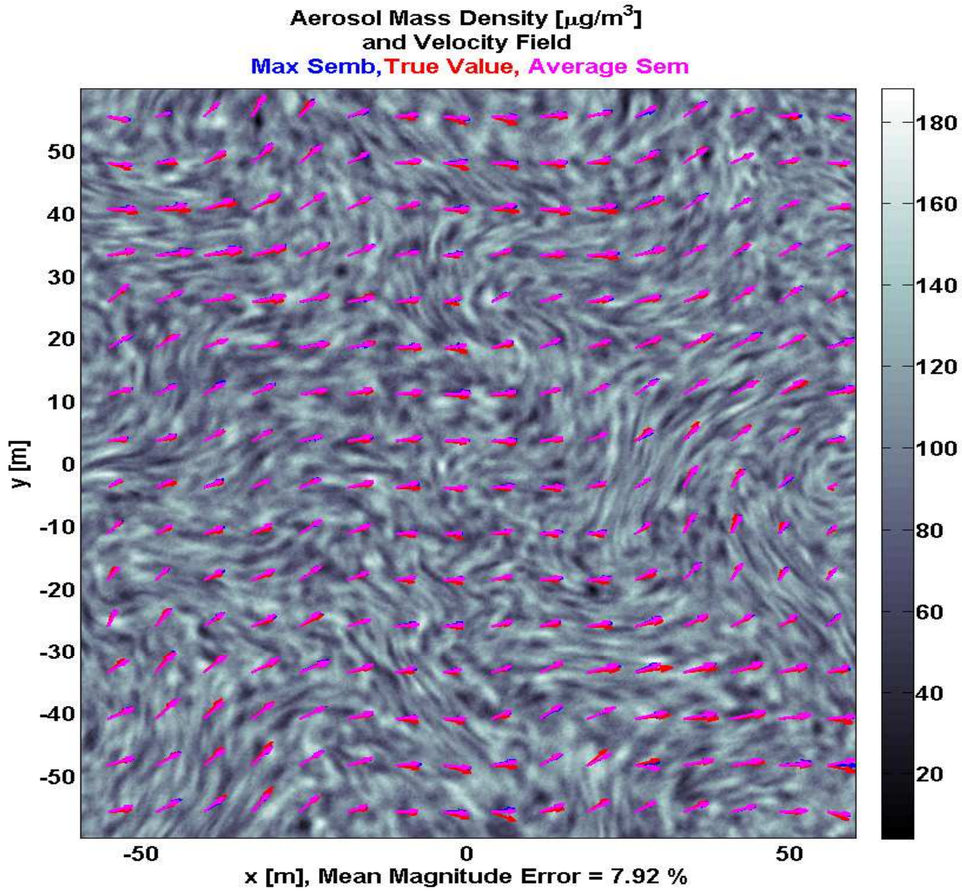
Table 2. Summary statistics for SM processing averaged over all 45 orbiting cloud images.

	$\mathrm{med}(\mathrm{v})$	$\mathrm{med}(\hat{\mathrm{v}}_{\mathrm{max}})$	$\mathrm{med}(\hat{\mathrm{v}}_{\mathrm{grd}})$	$\mathrm{avg}(\mathrm{v})$	$\mathrm{avg}(\hat{\mathrm{v}}_{\mathrm{max}})$	$\mathrm{avg}(\hat{\mathrm{v}}_{\mathrm{grd}})$	$\mathrm{std}(\mathrm{v})$	$\mathrm{std}(\hat{\mathrm{v}}_{\mathrm{max}})$	$\mathrm{std}(\hat{\mathrm{v}}_{\mathrm{grd}})$
v_x	–0.5551	1.5656	1.0568	–0.0059	1.6367	1.5229	10.1753	10.0554	10.0582
v_y	–1.0219	0.7828	0.5480	–0.2116	1.3877	1.3432	6.0165	5.8793	5.9780
$\lvert\mathbf{v}\rvert$	11.8414	11.3974	11.6369	11.4624	11.3674	11.4055	2.3109	2.8579	2.8310

13. Stochastic Cloud Model Results

A Matlab® [29] script based on Section 9 in two spatial dimensions and time produces realistic turbulent dynamic data. The size of the output arrays for $u_x(x, y, t)$, $u_y(x, y, t)$ and $\rho(x, y, t)$ are chosen to be $[640, 640, 320]$. Computations use FFTs that efficiently utilize compound integers of the form $640 = 5 \times 2^7$, $320 = 5 \times 2^6$). The inertial wavenumber cutoff in Equation (46) is chosen as $k_i = 1/(4\pi)$, corresponding to large eddies of approximately 20 m as seen in Figure 3. In Equation (48), $\sigma = 2.87$. Matlab® uses the maximally performant FFTW algorithm [49] and automatically detects real input arrays saving almost the theoretical factor of two in space and time. Comparison of our implementation of segmented STM and SM on the stochastic cloud and data model, as developed in Section 9, shows semblance to be slightly more accurate in estimating wind field magnitude and direction. We believe this is partially explained by the unique ability of semblance to analyze both amplitude and pattern. It is also more efficient by approximately a factor of 2 because of the FFT implementation. The 3D synthetic data computation takes 22 s of CPU time on an i7 Intel computer with 12 GB memory. Each 640 × 640 frame of motion retrieval analysis requires approximately 0.5 CPU seconds. This can be improved by more efficient 2D interpolation.

Figure 3. Segmented SM processing result for one frame of stochastic cloud model data. Color bar scale aerosol density units are $\mu g/m^3$.



The large scale slowly varying wind component $\mathbf{u}_\ell(\mathbf{x}, t)$ from Equation (32) is defined as

$$\begin{aligned} u_{x\ell}(t) &= 10\cos(\tfrac{1}{2}t/T)\,, \\ u_{y\ell}(t) &= 3.5\cos(\tfrac{1}{2}t/T + \pi/4)\,, \end{aligned} \tag{66}$$

where $T = 40$ s, the discrete has $N_\omega = 320$ components $t = [-T/2 + dt/2 : dt : T/2 - dt/2]$. The wind speed units in Equation (66) are m/s. In the following examples the subset of times t_n for

$64 \leq n \leq 128$ is chosen for both large and small wind components for the dynamical simulations. Figures 3 through 7 are based on the same data set. Figure 3 shows one of 320 representative output frames of 2D vector wind processing superimposed on bone colored aerosol density map in $\mu g/m^3$. Image processing converts each 640 × 640 spatial grid to a 20 × 20 segmentation. Each of the 400 resulting segments is analyzed by a 64 × 64 FFT for local semblance computation. Figure 3 has three arrow colors. Red is the true value, blue is the maximum semblance estimate and magenta gives the averaged semblance estimate. Maximum semblance uses the velocity based on the maximum semblance output. The more robust averaged estimate uses the largest 100 semblance outputs for each segment and computes a semblance weighted average. Color bar at right assigns gray scale to aerosol mass density. Visual inspection of Figure 3 shows the arrows for the most part overlapping. Video clip avi files of the turbulent model and vector field processing, corresponding to single frame results of Figures 3 and 4, are in file mar21_wind_avi_files.pdf for viewing in the same directory as the file for this paper.

Figure 4. Runge–Kutta streamline example from stochastic model using 25 start points shown in yellow. Cubic interpolation is used to smooth the results.

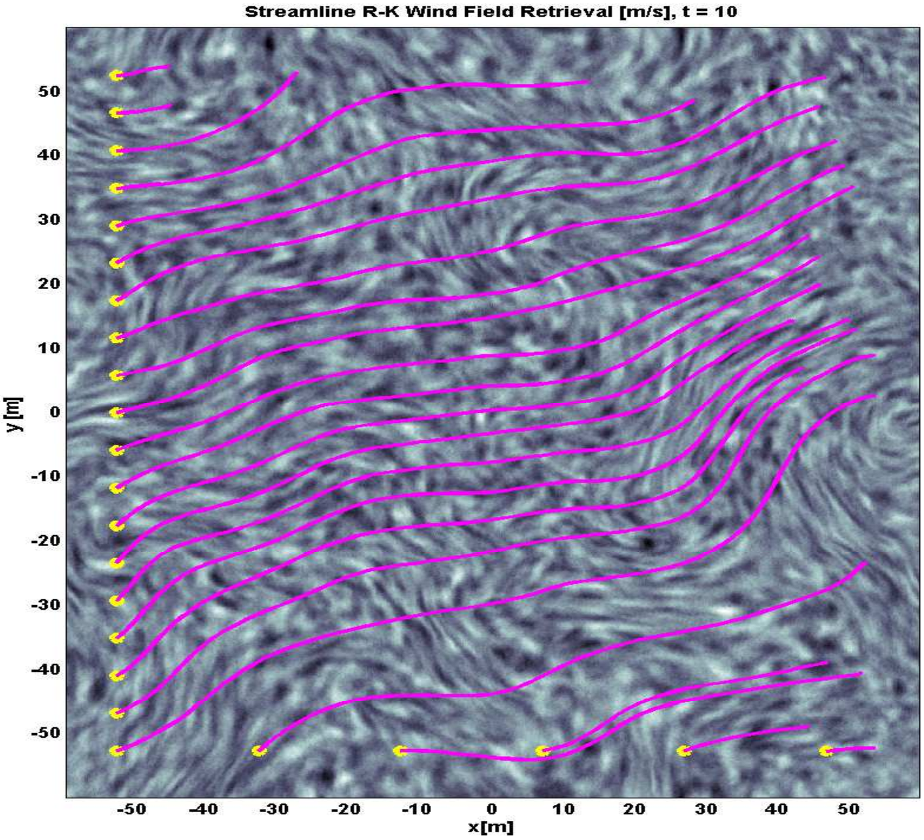
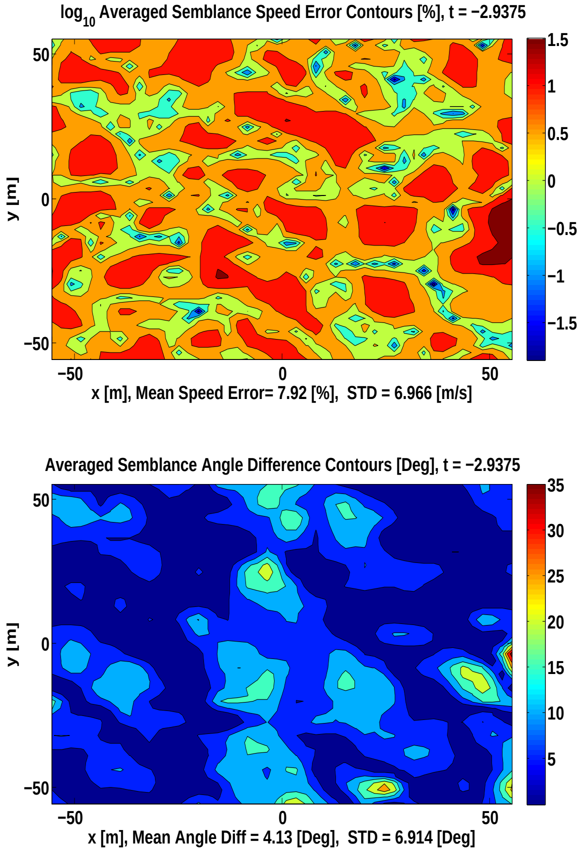


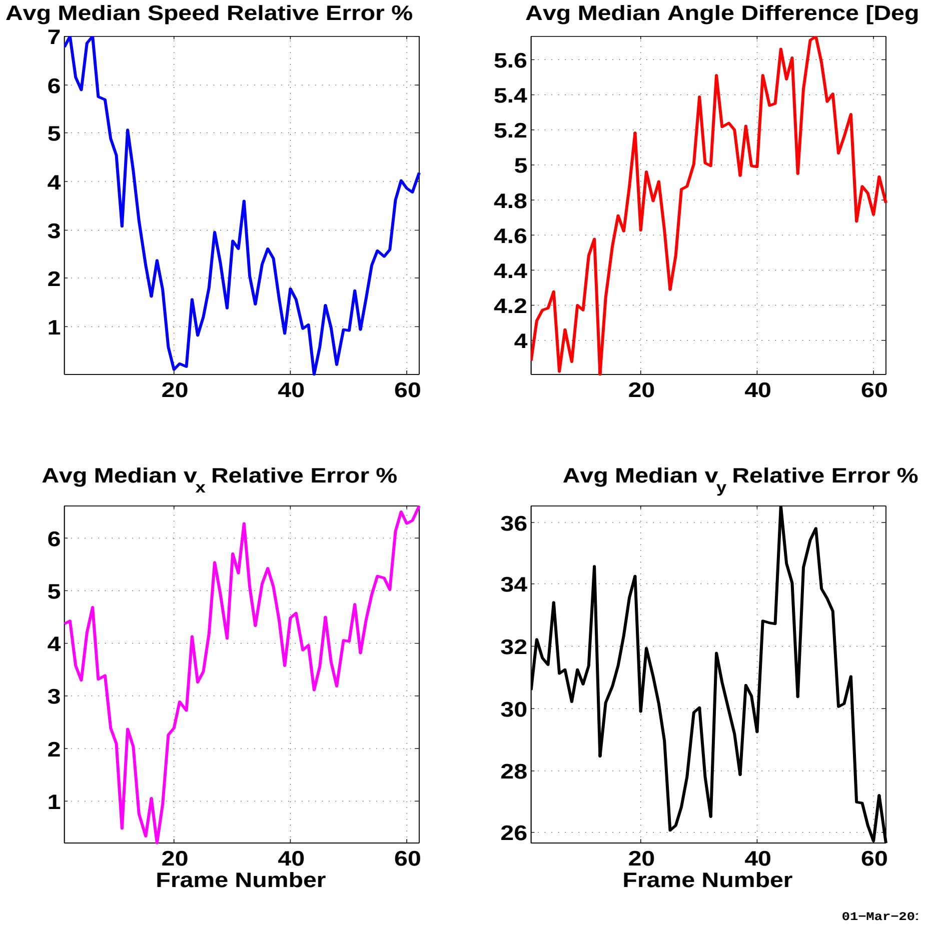
Figure 4 demonstrates the Runge–Kutta computation of streamlines. The processing selects 25 start points shown as yellow asterisks. The associated aerosol mass density is shown in background. The start points in this case are chosen to be uniformly spaced on left hand and bottom borders. Streamline smoothing uses five point cubic interpolation between each pair of Runge–Kutta points. The cubic sections use known end points and slopes. Figure 5 is a pixel by pixel color contour error analysis of Figure 3. Note the magnitude error has a $\log_{10}$ scale so that for example bright red corresponds to approximately 10% relative error with cooler colors less. Figure 6 is a frame-by-frame average median analysis of 64 successive time lapse image frames. Upper right hand subplot is difference in degrees between true and estimated median wind direction. Averaging over all frames of turbulent motion estimation, the average median speed error is approximately 7.5%. The average median pointing error is 5 degrees.

Figure 5. Associated magnitude and direction error analysis for Figure 3 data. Upper color bar units are $\log_{10}(p_e)$, where p_e is speed percent relative error. Lower color bar units are degrees.



The error analysis in Figures 5 and 6 do not tell the whole story. Time-lapse techniques are prone to yield biased mean speed estimates. Figure 7 plots frame-by-frame mean and median speed as well as component velocity statistics for 64 frames of time lapse data and compares them to the true values from the modeled wind field data. The bias for this more realistic turbulent model is evident and in these cases is less than ±1 m/s. The bias shifts from positive to negative during the 64 frame sequence. Such bias between anemometer and wind LIDAR are documented and explained in [1]. The idea is that wind LIDAR by necessity samples a larger volume of air than a cup anemometer to infer wind speed. Near the earth, there is a positive vertical gradient of wind speed due to surface drag that is more pronounced in areas with more terrain relief. Volumetric samples near the earth are therefore biased to smaller values than anemometers. The difference in measurement spatial averaging also explains the observed larger standard deviation of anemometer wind speeds in comparison to LIDAR. Anemometers measure smaller scale turbulence that the LIDAR measurement does not see. Methods for correcting for this bias are an area for future investigation.

Figure 6. Associated summary median percent relative error statistics for 64 time-lapse image frames.

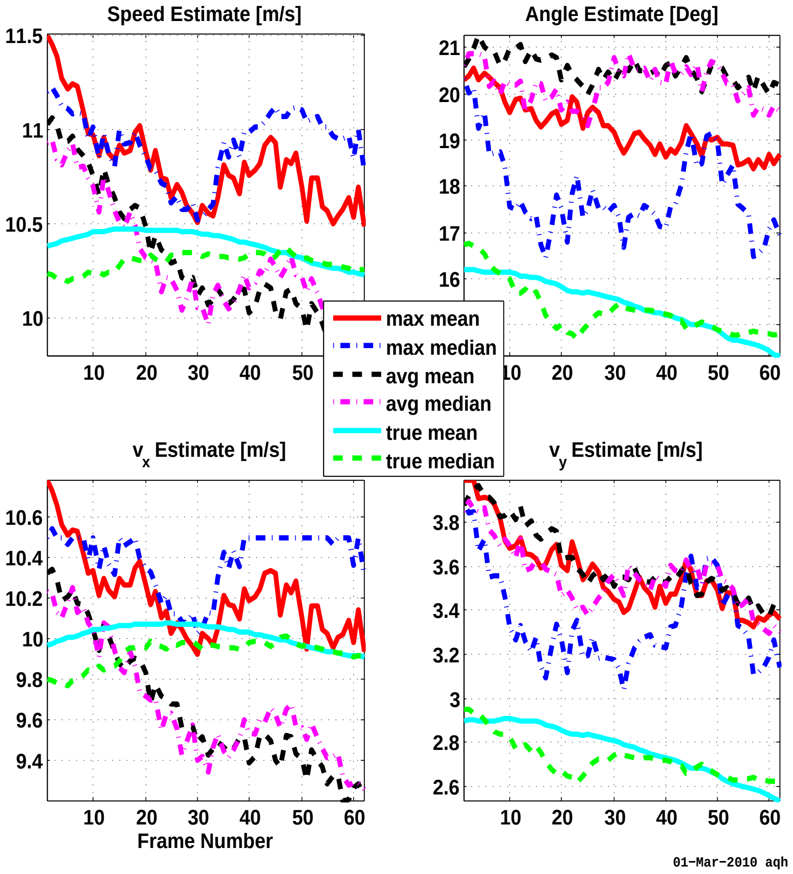


The techniques proposed here fail when the aerosol density is uniform, *i.e.*, when there are no measurable fluctuations. For example if the density is nearly constant in the vicinity of a flow line for a period of time in the volume data, velocity estimation in this vicinity by correlation is not possible. Doppler LIDAR wind and anemometers mapping remains useful in this case. Segmentation and the robust signal-to-noise-property of SM reduce, but cannot overcome, this deficiency.

Finally some words on integration of this process with hardware and applying it to 4D wind field estimation. Because such systems are monostatic, all signals are returned via a collecting telescope to the detector area in the order of 0.04 mm^2. Commercial 12 bit high-speed digitizers can now sample up to 2 GS/s. A cubic km of data at a resolution of 30 m^3 per voxel is 3.33×10^7 samples. Assume a 5 s period for the scanning system, and 100 sample averaging. This corresponds to 0.67 GS/s. The signal averaging can also be done on the digitizer card and then stored to a solid state drive with speeds up to 500 Mb/s. All of this is at the front end of the wind field estimation process outside of the CPU. The first step of the CPU process is to interpolate the 3D spherical coordinate data onto rectangular coordinates as described in the introduction. Linearly scaling this interpolation time to 3.33×10^7 samples yields 2 s. Because the data is 12 bit, faster integer arithmetic can be used. Also, as planes of rectangular data become available, the STM processing can begin in the populated planes. As expected, the majority of time is spent with the velocity field estimation step. For example, as presently coded, STM processing of 512 × 512 time lapse images takes 0.5 s. Linear scaling of this time to 3.33×10^7 voxels predicts an

STM processing time of 63 s for 1 km^3 of data at 30 m^3 voxel size resolution. Efficient coding can be expected to reduce this time by a factor of at least 2. Because the output of processing in parallel planes of data are independent, the STM process could then be implemented using 8 parallel threads to reduce the data cycle time to approximately 5 s.

Figure 7. Associated summary of frame mean and median statistics for 64 time-lapse image frames.



Another approach is to have a quick-look real-time process that outputs wind field data in several user defined slices through the volume data. The complete space time-data is stored for batch processing.

14. Summary and Conclusions

The trend in wind energy production is to use larger turbines requiring new techniques for wind field measurement. Scanning wind LIDARs are of two types: direct detection and coherent detection. Coherent detection uses relatively expensive transceivers with accurately frequency-modulated laser pulses and measures the Doppler frequency shift in the return signal, thus determining the radial velocity component. In comparison, direct detection aerosol backscatter LIDARs measure all three speed components equally well. This paper compares four methods for the general class of direct detection volume scanning LIDARs. The four methods are CCM, SM, TPSM and STM. The methods are first compared with a simple orbiting cloud model. The methods of SM and STM are found to be most accurate, with averaged mean-speed errors of $0.03\,\mathrm{m/s}$ for gradient weighted SM and $0.05\,\mathrm{m/s}$ for gradient weighted STM.

A second non-linear turbulent flow model provides for more realistic simulations and incorporates a FFT-PC based solution for the Navier–Stokes equation for isotropic and non-compressible flow. It does not contain terrain profile nor boundary layer effects but is nonetheless adequate for inter-comparison of the four retrieval methods. To simulate LIDAR backscatter data, an associated model for atmospheric aerosol density fluctuations $\rho(x, y, t)$ is implemented. Example results using this model demonstrate processing algorithm validation and comparison in a simulated environment where the true vector wind field is known. Extensions to the simulated wind field incorporating earth surface drag and boundary layer shear will result in more accurate turbulence scaling. Adding a more realistic saltation source to the aerosol transport model is another important extension. Correction for the wind speed bias through auxiliary measurement, extended modeling and/or image processing needs to be investigated. The way forward is to design, build, and evaluate a suitable, eye-safe, volumetric scanning backscatter LIDAR with effective range of at least 2 km, and spatial and time resolution respectively on the order 30 m^3 and 5 s.

Acknowledgments

The authors gratefully acknowledge USTAR funding and the support and collaboration of the CASI and EDL staff at Utah State University that made this work possible. The first author thanks the Geophysics Department of UFPa for their support during paper revisions. We also acknowledge the MDPI reviewers and editors for several important comments and suggestions leading to an improved paper.

References

1. Lang, S.; McKeogh, E. Lidar and sodar measurements of wind speed and directions in upland terrain for wind energy purposes. *Remote Sens.* **2011**, *3*, 1871–1903.
2. Sela, N. *The SpiDAR Wind Measurement Technique*; Available online: http://www.windtech-international.com/articles/the-spidar-wind-measurement-technique (accessed on 16 July 2012).
3. Kovalev, K.; Eichinger, W. *Elastic Lidar: Theory, Practice, and Analysis Methods*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2004.
4. Wilkerson, T.D.; Marchant, A.B.; Apedaile, T.J. Wind-field characterization from the trajectories of small balloons. *J. Atmos. Ocean. Technol.* **2012**, In Press.
5. Seinfeld, J.; Pandis, S. *Atmospheric Chemistry and Physics*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1998.
6. Eloranta, E.W.; King, J.M.; Weinman, J.A. The determination of wind speeds in the boundary layer by monostatic lidar. *J. Appl. Meteorol.* **1975**, *14*, 1485–1489.
7. Mayor, S.; Lowe, J.P.; Mauzey, C.F. Two-component horizontal aerosol motion vectors in the atmospheric surface layer from a cross-correlation algorithm applied to elastic backscatter lidar data. *J. Atmos. Ocean. Technol.* **2012**, In Press.

8. Hasager, C.B.; Badger, M.; Na, A.P.; Larsen, X.; Bingol, F. SAR-based wind resource statistics in the Baltic Sea. *Remote Sens.* **2011**, *3*, 117–144.
9. De Wekker, S.F.J.; Mayor, S. Observations of atmospheric structure and dynamics in the owens valley of California with a ground-based eye-safe, scanning aerosol lidar. *J. Appl. Meteorol. Climatol.* **2009**, *48*, 1483–1498.
10. Kunkel, K.E.; Eloranta, E.W.; Shipley, S.T. Lidar observations of the convective boundary layer. *J. Appl. Meteorol.* **1977**, *16*, 1306–1311.
11. Kunkel, K.E.; Eloranta, E.W.; Weinman, J.A. Remote determination of winds, turbulence spectra and energy dissipation rates in the boundary layer from lidar measurements. *J. Atmos. Sci.* **1980**, *37*, 978–985.
12. Piironen, A.K.; Eloranta, E.W. Accuracy analysis of wind profiles calculated from volume imaging lidar data. *J. Geophys. Res.* **1995**, *100*, 559–567.
13. Sroga, J.T.; Eloranta, E.W.; Barber, T. Lidar measurements of wind velocity profiles in the boundary layer. *J. Appl. Meteorol.* **1980**, *19*, 598–605.
14. Sasano, Y.; Hirohara, H.; Yamasaki, T.; Shimizu, H.; Takeuchi, N.; Kawamura, T. Horizontal wind vector determination from the displacement of aerosol distribution patterns observed by a scanning lidar. *J. Appl. Meteorol.* **1982**, *21*, 1516–1523.
15. Hooper, W.P.; Eloranta, E.W. Lidar measurements of wind in the planetary boundary layer: The method, accuracy and results from joint measurements with radiosonde and kytoon. *J. Appl. Meteorol.* **1986**, *25*, 990–1001.
16. Kolev, I.; Parvanov, O.; Kaprielov, B. Lidar determination of winds by aerosol inhomogeneities: Motion velocity in the planetary boundary layer. *Appl. Opt.* **1988**, *27*, 2524–2531.
17. Schols, J.L.; Eloranta, E.W. The calculation of area-averaged vertical profiles of the horizontal wind velocity from volume-imaging lidar data. *J. Geophys. Res.* **1992**, *97*, 18395–18407.
18. Mayor, S.; Eloranta, E.W. Two-dimensional vector wind fields from volume imaging lidar data. *J. Appl. Meteorol.* **2001**, *40*, 1331–1346.
19. Stam, J.; Fuime, E. Turbulent Wind Fields for Gaseous Phenomena. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH'93*, Anaheim, CA, USA, 2–6 August 1993; pp. 369–376.
20. Stam, J. A simple fluid solver based on the FFT. *J. Graph. Tools* **2001**, *6*, 43–52.
21. Sundquist, A. Dynamic line integral convolution for visualizing streamline evolution. *IEEE Trans. Vis. Comput. Graph.* **2003**, *9*, 273–282.
22. Weiskopf, D.; Eriebacher, G.; Ertl, T. A Texture-Based Framework for Spacetime-Coherent Visualization of Time-Dependent Vector Fields. In *Proceedings of IEEE Visualization 2003*, Seattle, WA, USA, 19–24 October 2003; pp. 107–114.
23. Marchant, A.B.; Barson, R.L.; Wojcik, M.; Howard, A.Q. Dynamic 3D Wind Mapping System and Method. US Patent 0149268 A1, 23 June 2011.
24. Oppenheim, A.; Schafer, R. Discrete-Time Signals and Systems. In *Digital Signal Processing*; Prentice Hall, Inc.: Englewood Cliffs, NJ, USA, 1975; pp. 242–244.
25. Tittman, J. The Physics of Logging Measurements. In *Geophysical Well Logging*; Academic Press, Inc.: Orlando, FL, USA, 1986; pp. 164–165.

26. Lim, J.S. Image Enhancement. In *Two-Dimensional Signal and Image Processing*; Prentice Hall PTR: Upper Saddle River, NJ, USA, 1990; pp. 503–506.
27. Tolman, R.C. Statistical Ensembles in the Classical Mechanics. In *The Principles of Statistical Mechanics*; Oxford University Press: Oxford, UK, 1962; pp. 43–54.
28. Golub, G.H.; Loan, C.V. Special Topics. In *Matrix Computations*, 2nd ed.; The John Hopkins University Press: Baltimore, MD, USA, 1989; pp. 579–633.
29. The Mathworks, Inc. *Matlab*®; Available online: www.mathworks.com (accessed on 2 August 2012).
30. Stull, R.B. Application of the Governing Equations to Turbulent Flow. In *An Introduction to Boundary Layer Meteorology*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1988; pp. 75–114.
31. Lesieur, M. Fourier Analysis of Homogeneous Turbulence. In *Turbulence in Fluids, Stochastic and Numerical Modelling*, 2nd ed.; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1990; pp. 137–138.
32. Kaimal, J.C.; Finnigan, J.J. *Atmospheric Boundary Layer Flows: Their Structure and Measurement*; Oxford University Press: New York, NY, USA, 1994.
33. Kaimal, J.C.; Wyngaard, J.C.; Izumi, Y.; Cote, O.R. Spectral characteristics of surface-layer turbulence. *Quart. J. Roy. Meteorol. Soc.* **1972**, *98*, 563–589.
34. Arfken, G. Vector Analysis. In *Mathematical Methods for Physicists*, 2nd ed.; Academic Press: New York, NY, USA, 1970; pp. 66–67.
35. Howard, A.Q., Jr. On the longitudinal component of the green's function Dyadic. *Proc. IEEE* **1974**, *62*, 1704–1705.
36. Knight, B.; Sirovich, L. Kolmogorov inertial range for inhomogeneous turbulent flows. *Phys. Rev. Lett.* **1990**, *65*, 1356–1359.
37. Bowman, J. On inertial-range scaling laws. *J. Fluid Mech.* **1996**, *306*, 167–181.
38. Lu, R.; Turco, R.; Jacobson, M. An integrated air pollution modeling system for urban and regional scales: I. Structure and performance. *J. Geophys. Res.* **1997**, *102*, 6063–6079.
39. Costabile, F.; Birmilli, W.; Klose, S.; Tuch, T.; Wehner, B.; Wiedensohler, A.; Franck, U.; Konig, K.; Sonntag, A. Spatio-temporal variability and principal components of particle number size distribution in an urban atmosphere. *Atmos. Chem. Phys.* **2009**, *9*, 3163–3195.
40. Stocker, T. *Introduction to Climate Change*; Springer: New York, NY, USA, 2011.
41. Brook, J.; Dann, T.; Burnett, R. The relationship among TSP, PM10, PM2.5, and inorganic constituents of atmospheric particulate matter at multiple Canadian locations. *J. Air Waste Manage. Assoc.* **1997**, *47*, 2–19.
42. Field, J.; Belnap, J.; Breshears, D.D.; Neff, J.C.; Okin, G.S.; Whicker, J.J.; Painter, T.H.; Ravi, S.; Reheis, M.C.; Reynolds, R.L. The ecology of dust. *Front. Ecol. Environ.* **2010**, *8*, 423–430.
43. Kok, J.F.; Renno, N.O. Electrostatics in wind-blown sand. *Phys. Rev. Lett.* **2008**, *100*, 014501.
44. Whitby, K.; Cantrell, B. Fine Particles. In *Proceedings of International Conference on Environmental Sensing and Assessments*, Las Vegas, NV, USA, 14–19 September 1975.
45. Shen, H.W.; Kao, D. A new line integral convolution algorithm for visualizing time-varying flow fields. *IEEE Trans. Vis. Comput. Graph.* **1998**, *4*, 98–108.

46. O'Neill, B. *Elementary Differential Geometry*; Academic Press: New York, NY, USA, 1966.
47. Moler, C.B. *Numerical Computing with Matlab*; SIAM: Philadelphia, PA, USA, 2004.
48. Brahms, N. *Fast 2-Dimensional Interpolation*; Available online: http://www.mathworks.com/matlabcentral/fileexchange/10772 (accessed on 31 May 2006).
49. Frigo, M.; Johnson, S. The design and implementation of FFTW3. *Proc. IEEE* **2005**, *93*, 213–216.