Estimating atmospheric motion winds from satellite image data using space-time drift models

Indranil Sahoo1 | Joseph Guinness2 | Brian J. Reich3

1Department of Statistical Sciences and Operations Research, Virginia Commonwealth University, Richmond, Virginia, USA
2Department of Statistics and Data Science, Cornell University, Ithaca, New York, USA
3Department of Statistics, North Carolina State University, Raleigh, North Carolina, USA

Correspondence
Indranil Sahoo, Department of Statistical Sciences and Operations Research, Virginia Commonwealth University, Richmond, VA, USA.
Email: sahooi@vcu.edu

Abstract
Geostationary weather satellites collect high-resolution data comprising a series of images. The Derived Motion Winds (DMW) Algorithm is commonly used to process these data and estimate atmospheric winds by tracking features in the images. However, the wind estimates from the DMW Algorithm are often missing and do not come with uncertainty measures. Also, the DMW Algorithm estimates can only be half-integers, since the algorithm requires the original and shifted data to be at the same locations, in order to calculate the displacement vector between them. This motivates us to statistically model wind motions as a spatial process drifting in time. Using a covariance function that depends on spatial and temporal lags and a drift parameter to capture the wind speed and wind direction, we estimate the parameters by local maximum likelihood. Our method allows us to compute standard errors of the local estimates, enabling spatial smoothing of the estimates using a Gaussian kernel weighted by the inverses of the estimated variances. We conduct extensive simulation studies to determine the situations where our method performs well. The proposed method is applied to the GOES-15 brightness temperature data over Colorado and reduces prediction error of brightness temperature compared to the DMW Algorithm.

KEYWORDS
asymmetry, derived motion winds, GOES-15, local maximum likelihood estimation, spatial smoothing, spatio-temporal processes

1 | INTRODUCTION

Wind field estimates are key inputs to many environmental studies. For example, winds are central to atmospheric circulation; therefore, the estimation of local or regional winds is important for making weather forecasts. Winds also influence vegetation as they affect factors that influence plant growth, such as seed dispersal rates, air transport of pollen, and metabolism rates in plants. The study of local winds also permits evaluation of power produced by wind turbines (Brown et al., 1984; Castino et al., 1998), prediction of propagation of oil-spills (Kim et al., 2014) and the study of coastal erosion (Ahmad et al., 2015). Local winds disperse air pollutants, for example when the Calima weather phenomenon blows dust into the Canary Islands (WeatherOnline, 2018). Winds also impact large-scale devastation such as forest fires.
For example, the Santa Ana winds blow into California after scorching summers (Berkowitz & Steckelberg, 2017; Bopp & Shaby, 2017) and produce conditions that lead to large wildland fires. Thus, it is important to map the strength and direction of local winds to help us prepare for natural calamities and facilitate preventive measures.

Observations of wind speed and direction at the ground level are collected at weather stations on land and by buoys or ships over oceans. There is a significant statistical literature on modeling winds from ground monitors (Brillinger, 2001; Haslett & Raftery, 1989; Priestley, 1981; Stein, 2005b). For some applications, it is sufficient to analyze the evolution of wind at a fixed location, and several methods have been proposed for this scenario (Ailliot, 2004; Brown et al., 1984; Monbet et al., 2007; Tol, 1997). The wind at different locations can be utilized to model spatio-temporal dependencies (Bennett, 1979; Bras & Rodriguez-Iturbe, 1985; De Luna & Genton, 2005; Kyriakidis & Journel, 1999). Also, Boukhanovsky et al. (2003), Malmberg et al. (2005) and Ailliot et al. (2006) proposed autoregressive space-time models to describe the evolution of winds. Stein (2005b) proposed a spectral-in-time modeling approach to describe the space-time dependencies of the data. Fuentes et al. (2008) modeled a drift process using Bayesian analysis, where the drift parameter is modeled using splines. Modlin et al. (2012) used circular conditional autoregressive models for wind direction and speed. Sigrist et al. (2012) modeled precipitation with a nonseparable spatio-temporal model using an external wind vector. However, these wind observations have very sparse coverage. More recently, spatial modeling of winds were done at annual (Jeong et al., 2018), monthly (Jeong et al., 2019) and daily (Tagle et al., 2019; Tagle et al., 2020) temporal resolutions. Also, Chen et al. (2021) developed a non-Gaussian space-time model to assimilate observed data and simulations to predict future winds.

Low orbit satellites such as Jason 3 and Sentinel 1 infer surface winds using geophysical inversion algorithms, based on peak backscattered power and the shape of radio signal waveforms (ESA, 2016). However, the satellites cannot monitor winds continuously in space and time, as they are in low earth orbit. Ground monitors can measure surface winds continuously in time but are sparse in space. Winds in the upper level of the atmosphere can be observed using weather balloons or aircraft measurements, but these observations are also very sparse in space and time.

On the other hand, geostationary weather satellites provide data from the surface and the atmosphere with a very high temporal resolution. The data comprise a series of images which essentially make them a “movie”. While the satellites do not directly measure wind, they measure infrared channel radiations in terms of brightness temperature. These brightness temperature image sequences are used to infer wind estimates by tracking movements of atmospheric tracers such as clouds or moisture features over time. Figure 1 shows a sequence of brightness temperature images captured by the Geostationary Operational Environmental Satellite (GOES)–15, operated by the National Oceanic and Atmospheric Administration (NOAA). Wind data obtained from satellite images play a major role in data assimilation (Lahoz & Schneider, 2014). Numerical climate models perform better with accurate wind data, especially over the oceans, resulting in improved weather forecasts and warnings (Tomassini et al., 1999). For example, the European Centre for Medium-Range Weather Forecasts (ECMWF) has been incorporating atmospheric motion winds into their forecast models operationally since the 1980s. This has dramatically improved the model’s ability to forecast the track of tropical cyclones and has also increased the model’s ability to predict wave heights and storm surges (Tomassini et al., 1999).

The Derived Motion Winds (DMW) Algorithm (Daniels et al., 2010) is a standard algorithm for estimating motion winds from satellite images. The DMW Algorithm takes as input brightness temperature images from the National Oceanic and Atmospheric Administration (NOAA) geostationary meteorological satellites and gives estimated wind fields as outputs. The algorithm tracks a suitable target across the input images and assigns a motion wind to the middle time point (see Section 2 for details). The image at the middle time point must satisfy a set of criteria to qualify as a suitable

Figure 1 A sequence of Brightness temperature (Kelvin) images over Colorado, captured by the GOES-15 satellite on January 3, 2015 at three consecutive time points.
As a result, the DMW estimates are often spatially and temporally sparse. The DMW Algorithm also does not provide a measure of uncertainty.

Geostatistical methods are capable of overcoming these shortcomings. This motivates us to model satellite image data using a spatial process drifting in time. At the heart of this statistical model lies the idea of incorporating the motion vector parameters in the process covariance. Stein et al. (2013) uses this idea to fit a space-time model. Instead of modeling the covariance function globally, we borrow the idea of Nested Tracking from Daniels et al. (2010) by considering local data buffers and estimate wind vectors using maximum likelihood estimates under the assumption of local stationarity. Local likelihood estimation of covariance parameters has been studied extensively in Haas (1990); Haas (1995); Ver Hoef et al. (2004); Risser and Calder (2015); Kuusela and Stein (2018); Lenzi et al. (2020); Wiens et al. (2020); Castro-Camilo and Huser (2020). Estimating covariance parameters locally helps tackle the nonstationarity present in the data. One major advantage of our approach is that it circumvents the computational burden for large sample sizes. Moreover, the proposed algorithm can easily be run in parallel, which can lead to further computational improvements. Another major advantage of the approach over the DMW Algorithm is that it allows us to quantify uncertainties in the local estimates of the winds; this in turn facilitates our use of an inverse-variance-weighted spatial smoothing algorithm to borrow strength across space while down-weighting the most uncertain estimates. This also allows tradeoffs between bias and variance for reducing estimation error (Anderes & Stein, 2011). The estimated motion winds obtained from the proposed model should be assigned at the height of the features being tracked (subject to availability of altitude data), which will vary over space and time. See the Conclusions section for a detailed discussion of this issue.

2 DERIVED MOTION WINDS ALGORITHM

The Derived Motion Winds (DMW) Algorithm estimates atmospheric motion winds from images taken by geostationary satellites. For a cloudy region, the imager records brightness temperature (see Figure 5), which measures the radiance (in Kelvin) of microwave radiation traveling upward from the top of the atmosphere to the satellite. For clear sky portions, the satellite records images of suitable indicators of atmospheric moisture content, such as specific humidity. Daniels et al. (2010) provides a description of and the physical basis for the estimation of atmospheric winds from the images taken by geostationary satellites.

The DMW Algorithm involves creating a data buffer, which is a data structure holding 2-dimensional arrays of brightness temperature for 3 consecutive image times. The middle portion of the buffer is divided into smaller target scenes (or target windows), and each scene is analyzed to locate and select a set of suitable targets in the middle image.

Daniels et al. (2010) also gives a description of the Nested Tracking Algorithm which involves nesting smaller target windows (usually of size $5 \times 5$) within a large target scene of size $15 \times 15$ pixels and getting every possible local motion vectors derived from each possible smaller window within a large target scene. The displacement vector between time points $t$ and $t + 1$ is computed by minimizing the Sum of Squared Differences (SSD) criterion as

$$\hat{v}(x, t, t + 1) = \arg\min_u \sum_{s \in D_x} (Y(s, t) - Y(s + u, t + 1))^2,$$

where $Y(s, t)$ denotes the brightness temperature within the smaller window at pixel location $s$ and time point $t$, $D_x$ is the indices of pixels in the target window centered at $x$, and $u$ is a two-dimensional vector denoting the displacement. The sum is considered over two dimensions and the optimization over $u$ is done only over integers so that $s + u$ corresponds to an observed pixel. In practice, the target scene is substantially larger than the size of the smaller target window, so the above summation is carried out for all target window positions within the target scene. The mean displacement vector is computed as

$$\hat{u}(x, t) = \frac{1}{2} \{\hat{v}(x, t - 1, t) + \hat{v}(x, t, t + 1)\}$$

and is assigned as the DMW estimate at location $x$ time point $t$ in the buffer. Once every possible local motion vector within the buffer are calculated, a density-based cluster analysis algorithm, DBSCAN (Ester et al., 1996) is used to identify the largest cluster representing the dominant motion. The final DMW estimate for the buffer is the average of the vectors belonging to the largest cluster.
The size of the target window depends on the spatial and temporal resolution of the imagery and the scale of the intended feature to be tracked. Daniels et al. (2010) suggests that the temporal resolution of the images should at most be 15 minutes in order to account for the short lifespan and rapid disintegration of clouds over land. The DMW Algorithm does not offer wind estimates at every space-time location as the data in the middle image has to satisfy a set of criteria to qualify as a suitable target scene (Daniels et al., 2010). Target scenes are processed to determine whether they (i) capture the intended target, (ii) contain sufficient contrast, and (iii) do not contain a mix of multiple potential targets. Target scenes that do not satisfy these criteria are deemed unsuitable for feature tracking and are rejected; no wind estimates are provided for those scenes. Also, quantifying uncertainties using the SSD criterion is hard because the subset of data used in the minimization criterion (1) depends on the parameter \( u \), so likelihood ratio tests are not applicable. Finally, the highest resolution of the vector estimates generated for each target scene can be half-integers.

3 | MODEL-BASED WIND ESTIMATION

3.1 | Space-time drift models

The proposed approach uses spatio-temporal covariance functions to track the wind. This requires us to consider asymmetric spatio-temporal covariance functions. Space-time asymmetries in covariance functions have been studied in Stein (2005a), Park and Fuentes (2006) and Huang and Sun (2019).

The space-time process \( Z(x, t) \) has asymmetric covariance if

\[
\text{Cov}[Z(x, t_1), Z(y, t_2)] \neq \text{Cov}[Z(x, t_2), Z(y, t_1)].
\] (2)

Here \((x, t)\) is a space-time location, with \( x = (x_1, x_2) \) giving the spatial coordinates, and \( t \) giving the time. In most regions, winds flow in a consistent direction, and so changes in brightness temperature and other atmospheric variables at one location tend to precede similar changes downwind. For instance, if \( t_2 > t_1 \) and winds flow consistently from \( x \) to \( y \), then we expect

\[
\text{Cov}[Z(x, t_1), Z(y, t_2)] > \text{Cov}[Z(x, t_2), Z(y, t_1)].
\]

We incorporate space-time asymmetries via a drift parameter. Suppose that \( Z_0 : \mathbb{R}^3 \rightarrow \mathbb{R} \) is a stationary, space-time symmetric process with covariance between \( Z_0(x, t_1) \) and \( Z_0(y, t_2) \) given by

\[
C_0((x, t_1), (y, t_2)) = M_v \left( \frac{[x_1 - y_1 + a_{12}(x_2 - y_2)]^2}{(1/a_{11})^2} + \frac{(x_2 - y_2)^2}{(1/a_{22})^2} + \frac{(t_1 - t_2)^2}{(1/a_{33})^2} \right)^{1/2},
\]

where \( M_v \) is the isotropic Matérn (Stein, 2012) covariance function,

\[
M_v(d) = \frac{\sigma^2}{2^{\nu-1} \Gamma(\nu)} k_{\nu}(d),
\]

where \( k_{\nu}(-) \) is the modified Bessel function of the second kind of order \( \nu \), \( \Gamma(-) \) is the gamma function, \( \sigma^2 \) is the spatial process variance, and \( \nu \) is the smoothness parameter which controls the mean square differentiability of the process. Also, \( a_{11} \) and \( a_{22} \) are the inverse spatial range parameters and \( a_{33} \) is the inverse temporal range parameter. \( Z_0 \) is geometrically anisotropic in its spatial coordinates with anisotropy parameter \( a_{12} \), inverse spatial ranges \( a_{11} \) and \( a_{12} \), and temporal range \( a_{33} \), but it is space-time symmetric because

\[
C_0((x_1, x_2, t_1), (y_1, y_2, t_2)) = C_0((y_1, y_2, t_1), (x_1, x_2, t_2)).
\]

Now, let us define

\[
Z(x, t) = Z_0(x + ut, t),
\]
where the two-dimensional vector \( u = (u_1, u_2) \) is the drift of the process over time and can be interpreted as the zonal and meridional components of wind. Intuitively, for a spatial process drifting from location \( x \) at time \( t_1 \) to location \( y \) at time \( t_2, t_2 > t_1 \), the optimal value of \( u \) can be thought of as the value that induces the highest correlation between the processes at \((x, t_1)\) and \((y, t_2)\). Then the covariance function of \( Z(x, t) \) is

\[
\text{Cov}[Z(x, t_1), Z(y, t_2)] = C_0 \{(x_1 + u_1 t_1, x_2 + u_2 t_1, t_1), (y_1 + u_1 t_2, y_2 + u_2 t_2, t_2)\}
\]

\[
= M_v \left( \left\{ \frac{[x_1 - y_1 + a_{12}(x_2 - y_2) + (u_1 + a_{12} u_2)(t_1 - t_2)]^2}{(1/a_{11})^2} + \frac{[x_2 - y_2 + u_2(t_1 - t_2)]^2}{(1/a_{22})^2} + (t_1 - t_2)^2 \right\}^{1/2} \right)
\]  

(3)

which makes the covariance function in Equation (3) space-time asymmetric and stationary. This model not only incorporates wind motions via the drift parameter, but it also explicitly models how brightness temperature (or other suitable tracer field) naturally evolves over time, even in the absence of winds.

### 3.2 Local estimation of the drift parameter

Assume the brightness temperature data \( Y(x, t) \) are normally distributed and have been standardized at each location (as described in the Appendix); denote the standardized data as \( Z(x, t) \). The mean and variance carry no information about the drift, and this step simplifies estimation of the correlation parameters. We do not specify a global covariance function for \( Z \). Instead, we specify its local covariance with drift parameter \( u(x, t) \in \mathbb{R}^2 \) and estimate the motion winds locally.

We define a target window as a square array of pixels

\[
D(x, t) = \{(x', t') : \|x - x'\|_\infty < \epsilon \& |t - t'| \leq 1\}.
\]

To estimate \( u(x, t) \), we assume that the process \( Z(x, t) \) is locally stationary in \( D(x, t) \) and that winds are smooth enough to be assumed constant in the scene, that is, \( u(x', t') \approx u(x, t) \) for all \( (x', t') \in D(x, t) \). In other words, we approximate the local covariance function as

\[
\text{Cov} \{Z(x_1, t_1), Z(x_2, t_2)\} \approx C_0 \{(x_1 + u(x, t)t_1, t_1), (x_2 + u(x, t)t_2, t_2)\}
\]

for \((x_1, t_1)\) and \((x_2, t_2)\) \in \(D(x, t)\).

We use maximum likelihood estimation in local, moving windows to make inference about the spatially varying covariance parameters \( \theta(x, t) = (\sigma^2(x, t), a_{12}^2(x, t), a_{22}^2(x, t), a_{12}^2(x, t), \nu(x, t), a_{12}(x, t), u(x, t)) \). The local estimation approach deals with the nonstationarity present in the entire data. We use maximum likelihood estimation within \( D(x, t) \) to estimate \( \theta(x, t) \equiv \theta_D \). If \( Z_D \) denotes the standardized data vector in the target window and \( \Sigma(\theta_D) \) denote the corresponding space-time covariance matrix with elements defined by (3), then the log-likelihood for \( \theta_D \) given \( Z_D \) is

\[
l(\theta_D|Z_D) = -\frac{1}{2} \log(|\Sigma(\theta_D)|) - \frac{1}{2} Z_D^T \Sigma^{-1}(\theta_D) Z_D.
\]

(4)

The estimated wind vectors are associated with the space-time location \((x, t)\) at which the target window \( D \) was centered, denoted \( \hat{u}(x, t) \). We also estimate the variances associated with the estimated wind vectors by computing the Fisher information at the maximum likelihood estimate, which gives a pointwise measure of uncertainty. We imitate the Nested Tracking approach and slide the target window across space and time, estimating wind vectors locally in space and time using the same optimization routine. In Section 4.1, we use the exact likelihood in (4) to estimate the wind vectors within a single target window. However, in Sections 4.2 and 5, we use a fast Vecchia approximation of the likelihood function (Guinness, 2018), implemented in the “GpGp” R package (Guinness & Katzfuss, 2018).

### 3.3 Smoothing the local estimates

After obtaining the local estimates of \( u(x, t) \) for all \((x, t)\), we smooth these initial estimates to stabilize them by borrowing strength across space. The two components of the wind vectors are smoothed separately. The kernel smoothing weights
are taken to be proportional to the ratio of a spatial Gaussian kernel and the variance of the initial estimate. Full details are given in the Appendix. The bandwidth is chosen based on cross validation.

The size of the target window is an important tuning parameter. While implementing the method on real data sets, the window size is chosen such that the wind motion is roughly constant in the scene, and the feature being tracked in time is prominent and does not move out of frame. In Section 4, we perform a simulation study analyzing the effect of the model parameters including window size on the performance of our method. We also analyze the efficacy of spatial smoothing of the local estimates. While implementing the methods in the real-data analysis in Section 5, the optimal window size is chosen using cross validation. This works well as wind fields are mostly smooth over a fairly small region and time frame.

### 4 | SIMULATION STUDIES

#### 4.1 | Motion wind estimation

In the first part of our simulation study, we determine the conditions under which the space-time drift model (STDM) performs well for estimating motion winds. For this purpose, we repeatedly simulate data sets within one particular target window (as opposed to scanning across a spatial domain) from a space-time Gaussian process with exponential covariance (obtained by substituting \( \nu = 1/2 \) and \( a_{12} = 0 \) in the expression for \( M_\nu \) in Equation (3)). We also implement a version of the DMW Algorithm and compare its performance with the STDM. To compare the two methods, accuracy for simulated data set \( i \) is measured by Vector Difference (Danielset al., 2010) between the true \( \mathbf{u}_0 \) and estimated \( \hat{\mathbf{u}}_i \)

\[
VD_i = \| \hat{\mathbf{u}}_i - \mathbf{u}_0 \|
\]

and we report the mean and standard deviation of \( VD_1, \ldots, VD_N \) over \( N = 100 \) data sets in Tables 1–3.

First we consider repeated data simulations in a target window of size \( 7 \times 7 \times 3 \), generated independently from a Gaussian process with covariance function \( M_\nu \) with \( \nu = 1/2 \) and \( \mathbf{u}(x, t) = \mathbf{u}_0 \) (constant true wind) using a fast Gaussian approximation algorithm (Guinness, 2018) which is implemented using the “GpGp” package (Guinness & Katzfuss, 2018) in R. We assume \( 1/a_{11} = 1/a_{22} = a_1 \) and the true spatial range parameter, \( a_1 \) is chosen to be either 1, 2, 4, or 8. The true temporal range parameter \( 1/a_{33} = a_2 \) is chosen to be either 1, 2, 3, or 4. We also take two different values of the true wind vector, namely \( \mathbf{u}_0 = (1, 2)^T \) and \( (3, 5)^T \), which signify respectively slow and fast wind vectors. Except \( \nu \), which is fixed at \( \nu = 1/2 \), all covariance parameters are updated simultaneously during optimization. Table 1 shows the performance of the two methods for the two wind vectors based on \( N = 100 \) simulations.

STDM does a better job in estimating moderately small and large wind vectors for the \( 7 \times 7 \) window size compared to DMW Algorithm. Mean vector distance is the smallest when the true spatial range is small and the true temporal range is large. This is intuitive because a small spatial range makes it easier to identify a feature in the target scene, and a large temporal range means that the features dissipate slowly over time. STDM also performs better for the smaller wind vector because when the wind vector is large compared to the window size, the feature tracked in time could potentially move out of the frame, resulting in incorrect wind estimates. In particular, windows of size \( 7 \times 7 \) are not adequate to contain the features being tracked in frame for a wind vector of \( (3, 5)^T \) which is reflected in the high MVD and SD values for STDM (see Table 1, top panel).

The performance of the DMW Algorithm also improves as the temporal range increases. However, these simulation results bring forth a major flaw in the DMW Algorithm. The estimated motion winds from the Algorithm are at most half integers and they are limited to the size of the larger search window. That is, while estimating motion vectors, the smaller central target scene \( (3 \times 3) \) can only move up to 4 pixels in all directions while it is being tracked back and forward in time. As a result, it performs poorly for the larger wind motion vector, which had a \( v \)-component of 5 pixel units. This can also be attributed to the window size relative to the magnitude of the wind vector.

To examine the effect of window size, we now perform the simulations again with window sizes of \( 11 \times 11 \) and \( 15 \times 15 \) respectively, under the same covariance parameter settings as described earlier. The panels of Tables 2 and 3 portray a clear picture of the effect of window size on the estimation of winds using STDM. The estimation of both wind vectors improve as we increase window size from \( 7 \times 7 \) to \( 11 \times 11 \) (see Table 2) and then to \( 15 \times 15 \) (see Table 3).
TABLE 1 Comparing Mean Vector Difference (SD) for Space-time Drift Model (STDM) and Derived Motion Winds Algorithm (DMWA) for true wind vectors \( \mathbf{u}_0 = (1, 2)^T \) (left panel) and \( \mathbf{u}_0 = (3, 5)^T \) (right panel) based on data window of size \( 7 \times 7 \); \( \alpha_1 \) and \( \alpha_2 \) denote the true spatial and temporal range respectively. MVD and SD are measured in the number of pixels.

| \( \alpha_1 \) \( \backslash \) \( \alpha_2 \) | 1         | 2         | 3         | 4         |
|----------------|----------|----------|----------|----------|
| 1              | 1.196 (0.35) | 0.311 (0.22) | 0.243 (0.24) | 0.201 (0.20) |
| 2              | 1.658 (1.13) | 0.600 (0.52) | 0.353 (0.33) | 0.245 (0.12) |
| 4              | 2.841 (1.75) | 1.599 (1.24) | 0.861 (0.61) | 0.456 (0.31) |
| 8              | 3.253 (2.25) | 3.281 (1.87) | 2.212 (1.71) | 1.432 (1.02) |

| \( \alpha_1 \) \( \backslash \) \( \alpha_2 \) | 1         | 2         | 3         | 4         |
|----------------|----------|----------|----------|----------|
| 1              | 1.983 (1.05) | 1.209 (0.85) | 1.039 (1.12) | 0.854 (1.09) |
| 2              | 2.045 (1.02) | 1.527 (0.94) | 1.283 (1.01) | 1.072 (0.98) |
| 4              | 2.342 (0.94) | 2.158 (0.94) | 1.620 (0.92) | 1.511 (0.91) |
| 8              | 2.452 (1.07) | 2.175 (0.95) | 2.003 (1.10) | 1.776 (0.89) |

| \( \alpha_1 \) \( \backslash \) \( \alpha_2 \) | 1         | 2         | 3         | 4         |
|----------------|----------|----------|----------|----------|
| 1              | 2.310 (1.60) | 2.034 (1.66) | 1.734 (1.68) | 1.787 (1.95) |
| 2              | 2.335 (1.56) | 2.032 (1.76) | 1.392 (1.47) | 1.030 (1.21) |
| 4              | 3.440 (2.34) | 2.393 (1.90) | 1.996 (1.70) | 1.638 (1.99) |
| 8              | 3.611 (2.66) | 3.135 (1.92) | 2.913 (1.88) | 2.483 (2.39) |

| \( \alpha_1 \) \( \backslash \) \( \alpha_2 \) | 1         | 2         | 3         | 4         |
|----------------|----------|----------|----------|----------|
| 1              | 5.953 (0.91) | 5.924 (0.96) | 5.953 (1.10) | 6.012 (0.98) |
| 2              | 5.710 (1.09) | 5.781 (1.18) | 5.899 (1.09) | 5.739 (1.08) |
| 4              | 5.665 (1.09) | 5.577 (1.16) | 5.540 (1.21) | 5.355 (1.13) |
| 8              | 5.952 (1.13) | 5.678 (1.23) | 5.413 (1.12) | 5.461 (1.18) |

In this part of the simulation study, the true wind has been chosen in a single window. This is analogous to constant wind vectors over a spatial domain. The next section of the simulation studies consider nonconstant wind fields and investigate the performance of the proposed method (including spatial smoothing) for different window sizes when data are generated from nonstationary/non-Gaussian processes. An additional simulation setting can be found in the Appendix.

4.2 Efficacy of spatial smoothing under different data generating models

We now conduct another simulation study which examines the performance of the competing methods and looks into the efficacy of spatial smoothing of wind estimates under three different data generation schemes. Unlike the simulation setting in Section 4.1, we now generate data sets on a \( 100 \times 100 \) unit square at three consecutive time points. While generating the data, we consider rotational winds, which are obtained by rotating each location counter-clockwise by \( 6^\circ \) around the origin at each time point. If \( \mathbf{x} \) denotes a spatial location at time \( t \) and \( \mathbf{x}' \) denotes the same at time \( (t + 1) \), then,

\[
\begin{pmatrix}
    x'_1 \\
    x'_2
\end{pmatrix} =
\begin{pmatrix}
    \cos \theta & -\sin \theta \\
    \sin \theta & \cos \theta
\end{pmatrix}
\begin{pmatrix}
    x_1 \\
    x_2
\end{pmatrix},
\]
TABLE 2 Comparing Mean Vector Difference (SD) for Space-time Drift Model (STDM) and Derived Motion Winds Algorithm (DMWA) for true wind vectors $\mathbf{u}_0 = (1, 2)^T$ (left panel) and $\mathbf{u}_0 = (3, 5)^T$ (right panel) based on data window of size $11 \times 11$; $\alpha_1$ and $\alpha_2$ denote the true spatial and temporal range respectively. MVD and SD are measured in the number of pixels.

| MVD of STDM for $\mathbf{u}_0 = (1, 2)^T$ |  |  |  |  |
|---|---|---|---|---|
| $\alpha_1 \setminus \alpha_2$ | 1 | 2 | 3 | 4 |
| 1 | 0.415 (0.37) | 0.172 (0.09) | 0.136 (0.07) | 0.118 (0.05) |
| 2 | 1.225 (1.21) | 0.304 (0.19) | 0.162 (0.09) | 0.149 (0.08) |
| 4 | 3.010 (2.56) | 1.008 (0.72) | 0.398 (0.25) | 0.268 (0.14) |
| 8 | 3.441 (3.48) | 2.830 (2.06) | 1.346 (1.00) | 0.831 (0.69) |

| MVD of DMWA for $\mathbf{u}_0 = (1, 2)^T$ |  |  |  |  |
|---|---|---|---|---|
| $\alpha_1 \setminus \alpha_2$ | 1 | 2 | 3 | 4 |
| 1 | 1.965 (1.17) | 0.771 (0.91) | 0.162 (0.50) | 0.084 (0.42) |
| 2 | 2.230 (1.30) | 1.343 (1.08) | 0.727 (0.82) | 0.409 (0.69) |
| 4 | 2.532 (1.32) | 1.888 (1.18) | 1.856 (1.17) | 1.101 (0.86) |
| 8 | 2.891 (1.54) | 2.538 (1.33) | 2.064 (1.09) | 1.961 (1.19) |

| MVD of STDM for $\mathbf{u}_0 = (3, 5)^T$ |  |  |  |  |
|---|---|---|---|---|
| $\alpha_1 \setminus \alpha_2$ | 1 | 2 | 3 | 4 |
| 1 | 1.124 (1.09) | 0.868 (1.50) | 0.743 (1.57) | 0.530 (1.06) |
| 2 | 1.916 (1.69) | 0.777 (1.24) | 0.299 (0.42) | 0.230 (0.41) |
| 4 | 2.820 (1.98) | 1.489 (1.36) | 0.676 (0.70) | 0.392 (0.37) |
| 8 | 3.401 (2.84) | 3.125 (1.96) | 2.071 (1.80) | 1.129 (1.03) |

| MVD of DMWA for $\mathbf{u}_0 = (3, 5)^T$ |  |  |  |  |
|---|---|---|---|---|
| $\alpha_1 \setminus \alpha_2$ | 1 | 2 | 3 | 4 |
| 1 | 5.823 (1.62) | 5.836 (1.43) | 5.866 (1.59) | 6.024 (1.61) |
| 2 | 5.638 (1.74) | 5.294 (1.75) | 5.408 (1.80) | 5.362 (1.64) |
| 4 | 5.548 (1.42) | 4.995 (1.65) | 4.934 (1.85) | 4.750 (1.70) |
| 8 | 5.864 (1.62) | 5.320 (1.72) | 5.233 (1.75) | 4.603 (1.70) |

for $t = 1, 2$ and $\theta = 6\pi/180$. Then

$$u_1(x, t) = x'_1 - x_1 \quad \text{and} \quad u_2(x, t) = x'_2 - x_2.$$ 

These winds are nonuniform over space and emulate big tropical storms which rotate counter-clockwise in the Northern Hemisphere and clockwise in the Southern Hemisphere due to the Coriolis effect. This simulation setting reflects realistic scenarios and compensate for the lack of availability of data sets with ground truth for wind measurements.

4.2.1 Data generating models

Under the first data generation scheme, the shifted locations at three consecutive time points are used to generate data sets from an approximate Gaussian process using the “GpGp” package (Guinness & Katzfuss, 2018) in R. The covariance function used for generating the data is the exponential space-time covariance ($M_v$ with $v = 1/2$) with true spatial range $\alpha_1 = 0.075$ (4 pixels), true temporal range $\alpha_2 = 5$ and a nugget variance of 0.1.
TABLE 3 Comparing Mean Vector Difference (SD) for Space-time Drift Model (STDM) and Derived Motion Winds Algorithm (DMWA) for true wind vectors \( \mathbf{u}_0 = (1, 2)^T \) (left panel) and \( \mathbf{u}_0 = (3, 5)^T \) (right panel) based on data window of size 15 × 15; \( \alpha_1 \) and \( \alpha_2 \) denote the true spatial and temporal range respectively. MVD and SD are measured in the number of pixels.

| \( \alpha_1 \backslash \alpha_2 \) | \( 1 \) | \( 2 \) | \( 3 \) | \( 4 \) |
|-----------------------------|-----|-----|-----|-----|
| \( 1 \)                 | 0.274 (0.32) | 0.125 (0.06) | 0.097 (0.05) | 0.073 (0.04) |
| \( 2 \)                 | 0.803 (0.71) | 0.197 (0.11) | 0.122 (0.07) | 0.103 (0.05) |
| \( 4 \)                 | 2.443 (1.64) | 0.536 (0.40) | 0.251 (0.14) | 0.183 (0.08) |
| \( 8 \)                 | 3.268 (3.60) | 2.030 (2.05) | 0.875 (0.59) | 0.496 (0.35) |

MVD of DMWA for \( \mathbf{u}_0 = (1, 2)^T \)

| \( \alpha_1 \backslash \alpha_2 \) | \( 1 \) | \( 2 \) | \( 3 \) | \( 4 \) |
|-----------------------------|-----|-----|-----|-----|
| \( 1 \)                 | 2.998 (1.55) | 1.501 (1.56) | 0.699 (1.33) | 0.318 (0.84) |
| \( 2 \)                 | 2.776 (1.42) | 2.028 (1.40) | 1.365 (1.57) | 0.815 (1.18) |
| \( 4 \)                 | 3.213 (1.69) | 2.661 (1.67) | 2.236 (1.67) | 1.644 (1.47) |
| \( 8 \)                 | 3.579 (1.70) | 2.921 (1.39) | 2.700 (1.53) | 2.518 (1.34) |

MVD of STDM for \( \mathbf{u}_0 = (3, 5)^T \)

| \( \alpha_1 \backslash \alpha_2 \) | \( 1 \) | \( 2 \) | \( 3 \) | \( 4 \) |
|-----------------------------|-----|-----|-----|-----|
| \( 1 \)                 | 0.581 (1.16) | 0.403 (1.21) | 0.355 (0.88) | 0.291 (0.99) |
| \( 2 \)                 | 1.073 (1.14) | 0.244 (0.13) | 0.142 (0.07) | 0.107 (0.06) |
| \( 4 \)                 | 2.781 (2.21) | 0.690 (0.65) | 0.307 (0.18) | 0.211 (0.11) |
| \( 8 \)                 | 3.175 (3.46) | 2.721 (2.55) | 1.123 (0.97) | 0.665 (0.53) |

MVD of DMWA for \( \mathbf{u}_0 = (3, 5)^T \)

| \( \alpha_1 \backslash \alpha_2 \) | \( 1 \) | \( 2 \) | \( 3 \) | \( 4 \) |
|-----------------------------|-----|-----|-----|-----|
| \( 1 \)                 | 5.100 (2.54) | 2.830 (2.46) | 0.688 (1.54) | 0.098 (0.62) |
| \( 2 \)                 | 5.306 (2.43) | 3.417 (2.51) | 1.819 (2.16) | 1.028 (1.77) |
| \( 4 \)                 | 5.724 (2.47) | 3.951 (2.42) | 3.378 (2.37) | 2.670 (2.32) |
| \( 8 \)                 | 6.109 (2.26) | 5.151 (2.44) | 4.393 (2.33) | 4.061 (2.56) |

The second data generation model is a Gaussian process model with nonstationary Matérn covariance function (Paciorek & Schervish, 2006) with nugget. In particular, we now assume that the spatial range parameter in the covariance function is spatially varying. The nonstationary covariance function has been defined at the shifted locations, so as to incorporate the rotational shift in the generated data. Analytically, the space-time version of the nonstationary Matérn covariance function can be written as

\[
C_{NS}\{(\mathbf{x}, t), (\mathbf{x}', t+1)\} = |\Sigma|^{-\frac{1}{2}} |\Sigma'|^{-\frac{1}{2}} \left( \sqrt{Q\{(\mathbf{x}, t), (\mathbf{x}', t+1)\}} \right) + \tau^2 I(\mathbf{x} = \mathbf{x}')
\]

with

\[
Q\{(\mathbf{x}, t), (\mathbf{x}', t+1)\} = [(\mathbf{x}, t) - (\mathbf{x}', t + 1)]^T \left( \frac{\Sigma + \Sigma'}{2} \right)^{-1} [(\mathbf{x}, t) - (\mathbf{x}', t + 1)],
\]

where \( \Sigma = \Sigma(\mathbf{x}) = \frac{\sigma^2}{2} I_2 \) and \( \Sigma' = \Sigma(\mathbf{x}') \) are kernel covariances which evolve over the spatial domain to produce nonstationary covariances. Here, \( I_2 \) denotes the identity matrix. For our simulation, we choose \( \sigma^2 = 2, \nu = 1.5 \) and \( \tau^2 = 0.1 \).
Finally, for our third simulation setting, we generate data from a non-Gaussian process, in particular, a Gamma process. For this purpose, we first generate Gaussian data \(Z(\mathbf{x}', t)\) at the shifted locations \(\mathbf{x}'\) for three consecutive time points, and then set

\[
Y(\mathbf{x}', t) = F^{-1}(\Phi(Z(\mathbf{x}', t)))
\]

where \(F\) is the CDF of a Gamma distribution with parameters \((a, b)\) and \(\Phi\) is the standard normal CDF. We choose \(a = b = 2\) in our simulation. Since we would never know if the data were exactly Gaussian, in practice, we often fit a Gaussian process to non-Gaussian data. Thus, the goal of this simulation setting is to evaluate the efficacy of the proposed local model under slight deviations from normality.

### 4.2.2 Evaluating model performance

From each of the three data generating schemes, we simulate \(N = 50\) data sets and fit the proposed space-time drift model locally. For data generated under the exponential space-time covariance model, we estimate all covariance parameters except the smoothness parameter, which is fixed at \(\nu = 0.5\). For the other two settings, all covariance parameters are simultaneously estimated. We estimate the parameters at \(75 \times 75\) locations at the center of the grid. This reduces boundary effects during local estimation and allows us to compare the performance of our method for multiple window sizes. The wind estimates are spatially smoothed using a weighted Gaussian kernel. We also estimate the winds using DMW Algorithm under similar settings, and the DMW estimates are also smoothed for fair comparison. For both methods, the optimal smoothing bandwidth has been chosen using cross-validation.

Table 4 compares the average performance of the algorithms and their smoothed versions over \(N = 50\) data sets, simulated from the three different settings. From the results, it can be seen that the STDM outperforms the DMW Algorithm for all window sizes, expect for a large window size like \(25 \times 25\) under the non-Gaussian setting for which the performances are comparable. Also, under every setting, smoothing the estimates gives us a much better representation of the wind fields compared to the raw estimates and STDM outperforms DMW Algorithm once smoothing is implemented. Also, Figures 2–4 show the respective true wind fields (in red) along with the raw and smoothed estimates (in blue) from both methods for a single representative data set from each of the three settings respectively, obtained using windows of size \(25 \times 25\). The wind vectors are plotted for a subset of the \(75^2\) locations in the spatial domain for clear visualization of the true and estimated fields. From the figures, it can be seen that the space-time drift model, fitted locally, efficiently captures the overall nonstationarity in the spatial fields and performs well under deviations from normality.

| Data generating model                     | Method      | 7 \times 7 | 11 \times 11 | 15 \times 15 | 21 \times 21 | 25 \times 25 |
|------------------------------------------|-------------|------------|--------------|--------------|--------------|--------------|
| Gaussian process with exponential space-time covariance | STDM        | 7.375 (6.007) | 5.357 (5.450) | 3.903 (4.745) | 2.414 (3.373) | 1.058 (1.854) |
|                                          | DMWA        | 7.551 (2.418) | 7.065 (3.056) | 6.143 (3.778) | 3.267 (3.440) | 1.371 (1.937) |
|                                          | Smoothed STDM | 6.158 (3.602) | 4.337 (3.925) | 2.659 (3.179) | 1.432 (1.733) | 0.571 (0.528) |
|                                          | Smoothed DMWA | 7.473 (2.156) | 6.912 (2.616) | 5.776 (2.958) | 2.711 (2.095) | 0.959 (0.735) |
| Gaussian process with nonstationary Matérn covariance | STDM        | 8.173 (11.038) | 3.995 (5.798) | 2.563 (3.975) | 1.474 (1.643) | 1.197 (1.109) |
|                                          | DMWA        | 7.388 (3.106) | 6.717 (3.294) | 5.498 (3.268) | 3.932 (3.074) | 2.733 (2.632) |
|                                          | Smoothed STDM | 5.792 (4.873) | 3.412 (3.822) | 2.040 (3.223) | 1.039 (0.562) | 0.935 (0.540) |
|                                          | Smoothed DMWA | 7.301 (2.975) | 6.554 (3.044) | 5.122 (2.734) | 3.403 (2.140) | 2.182 (1.749) |
| Non-Gaussian process                     | STDM        | 7.533 (6.048) | 6.010 (5.776) | 4.950 (5.773) | 3.122 (4.995) | 2.916 (4.418) |
|                                          | DMWA        | 7.817 (3.153) | 7.440 (3.554) | 6.704 (4.018) | 4.480 (4.147) | 2.556 (3.060) |
|                                          | Smoothed STDM | 5.954 (3.811) | 4.681 (4.221) | 3.435 (4.013) | 1.818 (2.731) | 1.396 (2.090) |
|                                          | Smoothed DMWA | 7.729 (3.003) | 7.267 (3.282) | 6.023 (3.370) | 3.845 (3.019) | 1.921 (1.731) |
FIGURE 2 True (red) and estimated (blue) wind fields using the Space-Time Drift Model (top row) and the DMW Algorithm (bottom row) for one representative dataset, simulated from a Gaussian process with the exponential space-time covariance function using rotational wind field. The left column shows raw wind estimates calculated using window size $25 \times 25$ and the right column shows the corresponding smoothed estimates. The wind vectors are plotted for a subset of spatial locations for clear visualization.

FIGURE 3 True (red) and estimated (blue) wind fields using the Space-Time Drift Model (top row) and the DMW Algorithm (bottom row) for one representative dataset, simulated from a Gaussian process with nonstationary covariance function using rotational wind field. The left column shows raw wind estimates calculated using window size $25 \times 25$ and the right column shows the corresponding smoothed estimates. The wind vectors are plotted for a subset of spatial locations for clear visualization.
The size of the target window plays a crucial role while estimating motion winds. For example, while estimating the rotational wind field using $15 \times 15$ (or smaller) windows, some of the target features move out of scene because of the nonuniformity in winds across space. In that scenario, irrespective of the data generating scenario, both methods provide poor estimates of the wind field. Using larger target windows, such as windows of size $25 \times 25$, fixes the problem to some extent. This corresponds to results from experimental studies carried out by special GOES-10 rapid scan imagery which showed that a higher number of high quality winds could be derived with decreasing time intervals and increasing spatial resolution (Daniels et al., 2010). For rotational wind field, larger window size improves estimation of the wind fields. This is because the average of the rotational wind field over a window is equal to the wind vector at the center of the window. Thus, a major takeaway from the simulation studies is that while estimating nonuniform wind fields, selecting the window size requires balancing a trade-off between a window size that is large enough to capture targets moving through the scene yet small enough to satisfy the assumption that within the window the process is stationary with a constant drift. Also, there is computational trade-off, since larger window sizes will lead to heavier computations. These motivate us to choose the window size using cross validation in the real-data analysis.

5 | APPLICATION TO GOES-15 DATA

5.1 | GOES-15 data description

The Geostationary Operational Environmental Satellite (GOES), operated by National Oceanic and Atmospheric Administration (NOAA) provides continual measurements of the atmosphere and surface variables, which help facilitate meteorological research including weather forecasting and severe storm tracking. Since the launch of GOES-8 in 1994, the GOES instruments have monitored atmospheric phenomena and provided a continuous stream of environmental data. The data set used in this project is from the GOES-15 satellite. Launched in March 2010, GOES-15 is positioned at the GOES-West location of $135^\circ$ W longitudes over the Pacific Ocean. The data set, as described in Knapp and Wilkins (2018), is a gridded satellite Contiguous US domain data which are geostationary data remapped to equal angle projection with an
FIGURE 5  Data buffers containing brightness temperature (Kelvin) maps over Colorado on January 3, 2015 at 00:15 am (top panel) and on January 4, 2015 at 08:45 am (bottom panel).

0.04° (~4 km) latitudinal resolution and 15 min temporal resolution. The data set includes infrared channel data in terms of pixel-wise brightness temperature for the reflective bands (channels 1–6 with approximate central wavelengths 0.47, 0.64, 0.865, 1.378, 1.61, 2.25 microns respectively). The reflective bands support among other ground and atmospheric indicators, the characterization of clouds. Gridded GOES-15 data can be obtained from NOAA One-Stop at https://data.noaa.gov/onestop/#/collections/details/AWbwYbuZHTVRBOAZkAzC?q=GOES.

We analyze Channel 4 brightness temperature data (recorded in Kelvin scale) for 10 consecutive days starting January 1, 2015 at a temporal resolution of 15 minutes (960 total images), covering the state of Colorado (36.82° N to 41.18° N latitudes and 109.78° W to 101.02° W longitudes) consisting of 6050 spatial locations at each time point. We choose Colorado as our study domain since, traditionally, Colorado has posed challenges to spatio-temporal modeling of atmospheric processes because of its diverse and complex terrain (Kleiber et al., 2013; Paciorek & Schervish, 2006). Central Colorado is home to the Rocky Mountains, which reaches elevations of over 4000 m. To the east of the Rockies, the terrain gradually transitions into the Great Plains, dipping to a minimum elevation of approximately 1000 m. Thus, meteorological processes evolving across the domain are likely nonstationary and cannot be modeled using simple isotropic covariance functions. Moreover, several studies have concluded that winds travel faster, and often turn into windstorms, as they flow along the downward slope of the Continental Divide (Mercer et al., 2008). Also, Colorado has historically experienced windstorms during early January, such as the large-magnitude wind gusts in 2007 (Mercer et al., 2008). Therefore, the chosen spatio-temporal domain ensures that the spatial process modeled is nonstationary and the wind fields to be estimated are nonuniform, which helps best illustrate the proposed method.

Figure 5 shows the data in two representative data buffers centered at 00:15 am on January 03, 2015 (t = 193; top panel) and at 08:45 am on January 04, 2015 (t = 322; bottom panel) respectively. These two time points represent wind movements across the state at night and in the morning respectively. Lower values of brightness temperature indicate cloud cover, whereas high brightness temperature values suggest clear skies over the region.

5.2 Estimation using GOES-15 data

The first step involves standardizing the brightness temperature data using the pixel-wise sample mean and standard deviation over time. This removes the constant background image and the effect of low cloud cover over the region which can affect the local estimation of wind vectors. To smooth the standard deviation map, we use a Gaussian kernel with smoothing parameter $\lambda = 3$ pixels (i.e., 12 km).
We fit a Gaussian process with the full covariance function in (3) to the standardized data within each target window using the ‘matern_anisotropic3D_alt’ covariance function in the “GpGp” package in R. As mentioned earlier, the covariance function in (3) includes an additional parameter $a_{12}$ to allow for anisotropy in the spatial covariances. All covariance parameters, including the smoothness parameter, are estimated locally within each window. We estimate the parameters at locations in the center of our spatial domain, so that the target windows are always a square around the target locations. We use cross validation to determine the appropriate size of target windows. Since no direct measurements of winds are available, we compare window sizes indirectly based on predictions of brightness temperature at the fourth time point using the wind estimate based on the first three time points and use this optimal window size to estimate winds for all subsequent time points. Table 5 gives the Mean Squared Prediction Error (MSPE) and the Continuous Rank Probability Score (CRPS) (discussed in Section 5.3) for different window sizes and the corresponding average computation time (in seconds) to estimate the wind vector at one pixel using 3 consecutive time steps. All computations are done on a PC with Intel Core i7-9750H CPU at 2.60GHz with 32 GB of RAM. Based on a lower cross validation MSPE, comparable CRPS and a much lower computation time, we select subregions of size $25 \times 25$ pixels (i.e., $100 \text{ km} \times 100 \text{ km}$) and estimate the wind vectors, along with all other covariance parameters, locally at each spatial location and at each time using maximum likelihood estimates. We also estimate the variances associated with the estimates by computing the Fisher information at the MLE. Figure 6 shows the uncertainty associated with the estimated wind components as given by the estimated standard deviations in the log scale, for the two data buffers mentioned above.

The estimated variances are high at some locations, especially where it is difficult to distinguish clouds from the background terrain and track their movements across the region. To account for this, we smooth each component of the estimated wind field using weighted Gaussian kernels, the weights being scaled to the inverse variances of the estimates (details are given in the Appendix). The smoothing parameter has been chosen based on cross validation.

Similar to the DMW Algorithm, we also impose some simple criteria to check whether a target window $D(x, t)$ is suitable for tracking and estimating motion winds. We look at the mean of the residual variances from regressing $Z_D(\cdot, t)$ on $Z_D(\cdot, t - 1)$ and $Z_D(\cdot, t + 1)$ on $Z_D(\cdot, t)$. We say that $D(x, t)$ is discarded if the mean residual variance is low and the brightness temperature at the location, $Y(x, t)$ is high. This signifies no cloud cover or negligible movement across time within the target window. For our data, we reject $D(x, t)$ if the mean residual variance is in the lower 10 percentile of all mean residual variances and the brightness temperature $Y(x, t)$ is in its upper 10 percentile, which is, $Y(x, t) > 260$ Kelvin. Figure 7 shows the raw and smoothed wind estimates obtained from the proposed STDM. The wind vectors have been plotted at a subset of spatial locations for better visualization.

For comparison, the DMW estimates are also smoothed using a simple Gaussian kernel smoother with its optimal smoothing parameter chosen using cross validation. The raw and smoothed estimates of the wind fields obtained from the DMW Algorithm are plotted at a subset of locations in Figure 8. The results once again highlight the limitations of the DMW Algorithm output. While the space-time drift model is able to capture the consistent winds over Colorado, the DMW Algorithm does not identify the pattern. As a result, the wind fields from the DMW Algorithm look jumbled after smoothing.

### 5.3 Comparison based on Mean Squared Prediction Error and Continuous Rank Probability Score

For this dataset, reference wind fields are not available. Hence, we compare the two methods based on Mean Squared Prediction Error (MSPE) and Continuous Rank Probability Score (CRPS; see (Gneiting & Katzfuss, 2014;
Estimated log SD for $\vec{u}_1$ (Jan 3; 00:15 am)

Estimated log SD for $\vec{u}_2$ (Jan 3; 00:15 am)

(a) Estimated log SD for $\vec{u}_1$ (Jan 4; 08:45 am)

(b) Estimated log SD for $\vec{u}_1$ (Jan 4; 08:45 am)

(c) Estimated log SD for $\vec{u}_2$ (Jan 3; 00:15 am)

(d) Estimated log SD for $\vec{u}_2$ (Jan 4; 08:45 am)

FIGURE 6 Estimated standard deviations (in log scale, unit: log Kelvin) obtained using the Space-Time Drift Model (STDM), corresponding to estimated east-west (top row) and north-south (bottom row) wind components on January 3, 2015 at 00:15 am (left column) and January 4, 2015 at 08:45 am (right column) respectively.

Gneiting & Raftery, 2007) for details) while predicting standardized brightness temperature fields. To predict $Z$ at a spatial location $x$ at time $t$, we consider $Z_D(\cdot, t-1)$, the standardized data in $D(x, t-1)$ and estimated winds at time $t-2$. This is because the estimated winds at time $t-2$ uses data from times $t-3, t-2$ and $t-1$. The predicted standardized temperature is calculated as the mean of the Gaussian conditional distribution of $Z(s, t)$ given $Z_D(\cdot, t-1)$ under the stationary Matérn space-time drift model with estimated drift parameter $\hat{u}(s, t-2)$. We predict standardized brightness temperature using the raw and smoothed versions of the wind estimates from both algorithms. We also consider a naive approach of predicting the brightness temperature fields, where the data at time $t-1$ is considered to be the predicted standardized brightness temperature fields at time $t$. We call it the persistence prediction and the performances of the two methods are assessed relative to the baseline MSPE. To calculate the CRPS, we assume the associated predictive distribution of standardized brightness temperature is well approximated by a Gaussian distribution with mean centered at the predicted value. We are also able to compute CRPS for predictions based on motion winds derived from DMW Algorithm by predicting standardized brightness temperature using the proposed probabilistic model. While making these predictions for DMW Algorithm, we use all estimated covariance parameters from the STDM, except the drift parameters. Table 6 compares the raw and smoothed wind estimates in terms of MSPE and CRPS for 4 consecutive time points corresponding to the two data buffers. From Table 6, we also see a moderate gain in prediction accuracy for these time points while predicting using the smoothed wind estimates, suggesting that local smoothing is preferred. Figure 9 provides maps of predicted
Figure 7  Initial (top row) and smoothed (bottom row) wind field estimates (km/15 mins) over Colorado obtained using the Space-Time Drift Model (STDM). The left column shows estimated wind on January 3, 2015 at 00:15 am and the right column shows the same on January 4, 2015 at 08:45 am. The wind vectors are plotted for a subset of spatial locations for clear visualization.

brightness temperature at a subset of spatial locations and at $t = 196$ and $t = 325$ respectively using the smoothed estimated winds from STDM. It can be seen that our model captures the main features in the brightness temperature fields over Colorado. Our model also captures the wind movement over the region since we can see similar patterns of features across the region as compared to the ones tracked along in the original images (see Figure 5).

6 | DISCUSSIONS AND CONCLUSIONS

6.1 | Discussion

Wind is one of the most important atmospheric variables; it has large impact on local weather and hence, studying winds is essential. In this paper, we propose a spatio-temporal model to derive atmospheric motion winds data from a sequence of high-resolution of images over time collected by geostationary satellites. In particular, we use a sequence of brightness temperature images taken by the GOES series of the NOAA meteorological satellites and estimate atmospheric wind speed and direction. We model brightness temperature using a space-time drift process and propose local estimation of
drift parameters. Developing a globally valid nonstationary space-time drift model is an interesting problem that we have not pursued here. Following the basic paradigm of the Nested Tracking implementation of the Derived Motion Winds Algorithm (DMWA), we propose local likelihood estimation of the covariance parameters. We smooth the raw estimated wind fields using a weighted Gaussian kernel, the weights being scaled by the inverse of the estimated variances of the estimates. We also compare the performance of our method with the DMW Algorithm.

Section 4.1 details extensive simulation studies that outlines conditions under which our model performs well. Based on our simulation study, we conclude that we have accurate wind estimates when the true spatial correlation range is small and the true temporal correlation range is high. The simulations in Section 4.2 show that our method performs well even when we have nonstationary and non-Gaussian spatial fields. The simulations also show that spatial smoothing provides better estimates of the wind fields. The simulations highlight a major drawback of the DMW Algorithm. Due to the design of the DMW Algorithm, the local DMW Algorithm estimates can only be half-integers and can only take values equal to the number of pixels the smaller target scene can move around in the larger search window. This limitation of the DMW Algorithm stems from the fact that the algorithm requires the original and shifted brightness temperature data to be at the same locations, in order to calculate the sum of squares criterion between the two images. However, this is not a requirement for our method, since we estimate the drift by maximizing a multivariate normal likelihood over a model that involves a continuous parameter $\mathbf{u}$, rather than relying on the sum of squared differences between the original and
TABLE 6 Comparing Mean Squared Prediction Error (standardized) and average CRPS (in parentheses) based on prediction using the raw and smoothed wind estimates from the Space-Time Drift Model (STDM) and the Derived Motion Winds Algorithm (DMWA) at different time points. $\lambda$ in each case denotes the optimal smoothing parameter chosen using cross validation. All the methods have been compared against the baseline prediction.

| Method               | $t = 196$ | $t = 197$ | $t = 198$ | $t = 199$ |
|----------------------|-----------|-----------|-----------|-----------|
| STDM                 | 0.329 (0.36) | 0.316 (0.35) | 0.404 (0.38) | 0.732 (0.49) |
| DMWA                 | 0.604 (0.44) | 0.561 (0.43) | 0.657 (0.46) | 0.985 (0.56) |
| Smoothed STDM ($\lambda = 2$ km) | 0.323 (0.34) | 0.300 (0.33) | 0.386 (0.37) | 0.716 (0.47) |
| Smoothed DMWA ($\lambda = 8$ km) | 0.486 (0.39) | 0.454 (0.38) | 0.553 (0.42) | 0.899 (0.52) |
| Persistence          | 0.596     | 0.568     | 0.614     | 1.137     |

| Method               | $t = 325$ | $t = 326$ | $t = 327$ | $t = 328$ |
|----------------------|-----------|-----------|-----------|-----------|
| STDM                 | 0.372 (0.37) | 0.588 (0.45) | 0.334 (0.36) | 0.684 (0.48) |
| DMWA                 | 0.580 (0.44) | 0.724 (0.49) | 0.582 (0.43) | 0.863 (0.52) |
| Smoothed STDM ($\lambda = 2$ km) | 0.370 (0.36) | 0.580 (0.44) | 0.324 (0.34) | 0.619 (0.47) |
| Smoothed DMWA ($\lambda = 8$ km) | 0.505 (0.41) | 0.664 (0.47) | 0.493 (0.40) | 0.776 (0.49) |
| Persistence          | 0.684     | 1.204     | 0.613     | 1.180     |

FIGURE 9 Predicted brightness temperature fields (Kelvin) at $t = 196$ and $t = 325$ obtained using corresponding smoothed wind estimates from the Space-Time Drift Model (STDM). The predictions have been made at a subset of spatial locations.

shifted images. This brings us to perhaps the most important tuning parameter in the analysis, the target window size. We show that the window size is very crucial for both methods with large bias resulting from a large window and variance resulting from a small window.

We apply our method on brightness temperature data over Colorado, obtained from the GOES-15 satellite. While estimating winds, the window size has been chosen using cross validation. We provide estimated standard deviation maps, showing that our method is capable of quantifying uncertainties associated with the estimation. Since the uncertainties in estimated wind speed are correlated over space and time and hence hard to estimate, we provide point-wise uncertainties. We also provide smoothed maps of estimated wind fields over Colorado. We use cross-validation to choose the smoothing bandwidth thereby implementing a data-driven method for how much smoothing to do. Since zonal (horizontal) and meridional (vertical) wind components are usually studied separately in various applications, we smooth the two components of the wind vector separately. However, one can smooth both wind components together using a bivariate Gaussian smoothing filter, thereby using the correlation between the estimated wind components, along with the estimated variances. We predict brightness temperature fields using our model and compare the raw and smoothed estimates of wind fields based on Mean Squared Prediction Error (MSPE) and Continuous Rank Probability Score (CRPS). We also
compare the performance of our proposed method and the DMW Algorithm. We have shown that our method outperforms the DMW Algorithm with respect to MSPE and CRPS. We argue that smoothing the estimated wind fields give more reliable wind estimates. We also see that we capture the main features in the brightness temperature maps through our prediction, including the drift across the region.

The estimated motion winds obtained from the DMW Algorithm are used as one of the input variables in the Multi-platform Tropical Cyclone Surface Wind Analysis (MTCSWA) model (Knaff et al., 2011), whose main purpose is to create an algorithm package to operationally generate an estimation of surface wind fields around active tropical cyclones. It has also been seen that, considering uncertainties of parameters usually results in improved projections from these large-scale scientific models (Sharma et al., 2023). As such, more accurate estimates of motion winds along with their estimated uncertainties obtained from the proposed method, can potentially improve the forecasts obtained from the MTCSWA model.

6.2 Conclusions

Wind speeds tend to increase with height above ground and as a result, wind speeds are usually measured at varying altitudes. According to Daniels et al. (2010), the GOES-R ABI cloud height algorithm is usually used to assign a representative height to each cloudy target. This algorithm uses National Centers for Environmental Prediction (NCEP) Global Forecast System (GFS) forecast temperature profiles and cloud-top pressure as ancillary data to assign a representative height to clouds in a target scene. However, since neither the output from the GOES-R ABI cloud height algorithm nor the ancillary data sets were available to us, we were unable to assign appropriate altitude levels to the estimated motion winds. If the heights were available, though, we would simply assign our estimated winds to the inferred top height of the clouds. Integrating these ancillary data sets to estimate the altitudes of the motion winds is nontrivial, and can be explored in a future project. The unavailability of data on cloud heights does make comparisons of estimated winds against like wind observations more difficult. Hence, we devised a validation scheme based on predicting future brightness temperatures based on the estimated wind fields.

The DMW Algorithm rejects target scenes that do not satisfy certain criteria and does not provide any wind estimates for those scenes. In our data application, we have tried to emulate the general setting of the Nested Tracking implementation of the DMW Algorithm, and have applied one of their target selection criterion on our data set to make the algorithms as comparable as possible. If the goal of the study is simply to report estimated winds when suitable tracer fields are available, then we think that the mechanism for missingness is not important. However, if one wants to use the estimated winds to infer long-term wind climatologies, then one should investigate the missingness mechanism. Both our method and the competitor are more likely to be missing on days without suitable tracers, and so if there is an association between cloud cover and wind then this is a source of informative missingness that should be taken into account when inputting missing values. We expect such relationships to be weak, but further study is warranted for long-term climatological inference.

One major challenge of our method is to apply it to data in real time. The main computational bottleneck is the time required for optimization over the covariance parameters. Because we use a local likelihood approach, the method is embarrassingly parallel across pixels and time points, and thus should scale well when used in production. The optimization can also be made faster using approximation. For example, the optimization algorithm could be initialized using the results of spatial or temporal neighbors, and the parameters could be updated using a one-step Fisher’s scoring approximation. We would also like to explore more flexible methods for capturing the complicated wind motions, possibly a method where the window size is allowed to vary with location and time. This can ensure that our feature is always inside the frame of reference, resulting in more accurate estimates of the wind fields.

ORCID

Indranil Sahoo © https://orcid.org/0000-0002-4960-7984

REFERENCES

Ahmad, N., Suzuki, T., & Banno, M. (2015). Analyses of shoreline retreat by peak storms using Hasaki coast Japan data. Procedia Engineering, 116, 575–582.

Ailliot, P. (2004). Modeles autorégressifs à changement de régimes Markovien-Application a la simulation du vent. PhD Thesis. Université de Rennes 1.
Mercer, A. E., Richman, M. B., Bluestein, H. B., & Brown, J. M. (2008). Statistical modeling of downslope windstorms in boulder, Colorado. *Weather and Forecasting*, 23, 1176–1194.

Modlin, D., Fuentes, M., & Reich, B. J. (2012). Circular conditional autoregressive modeling of vector fields. *Environmetrics*, 23, 46–53.

Monbet, V., Ailliot, P., & Prevosto, M. (2007). Survey of stochastic models for wind and sea state time series. *Probabilistic Engineering Mechanics*, 22, 113–126.

Paciorek, C. J., & Schervish, M. J. (2006). Spatial modeling using a new class of nonstationary covariance functions. *Environmetrics: The Official Journal of the International Environmetrics Society*, 17, 483–506.

Park, M. S., & Fuentes, M. (2006). *New classes of asymmetric spatial-temporal covariance models*. Tech. rep. North Carolina State University, Dept. of Statistics.

Priestley, M. B. (1981). *Spectral analysis and time series*. Academic Press London.

Risser, M. D., & Calder, C. A. (2015). Local likelihood estimation for covariance functions with spatially-varying parameters: The convospat package for R. arXiv preprint arXiv:1507.08613.

Sharma, S., Lee, B. S., Hosseini-Shakib, I., Haran, M., & Keller, K. (2023). Neglecting model parametric uncertainty can drastically underestimate flood risks. *Earth’s Future*, 11, e2022EF003050.

Sigrist, F., Künsch, H. R., & Stahel, W. A. (2012). A dynamic nonstationary spatio-temporal model for short term prediction of precipitation. *The Annals of Applied Statistics*, 6, 1452–1477.

Stein, M. L. (2005a). *Space–time covariance functions*. *Journal of the American Statistical Association*, 100, 310–321.

Stein, M. L. (2005b). Statistical methods for regular monitoring data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67, 667–687.

Stein, M. L. (2012). *Interpolation of spatial data: Some theory for kriging*. Springer Science & Business Media.

Stein, M. L., Chen, J., & Anitescu, M. (2013). Stochastic approximation of score functions for Gaussian processes. *The Annals of Applied Statistics*, 7, 1162–1191.

Tagle, F., Castruccio, S., Crippa, P., & Genton, M. G. (2019). A non-Gaussian spatio-temporal model for daily wind speeds based on a multi-variate skew-t distribution. *Journal of Time Series Analysis*, 40, 312–326.

Tol, R. S. (1997). Autoregressive conditional heteroscedasticity in daily wind speed measurements. *Theoretical and Applied Climatology*, 56, 113–122.

Tomassini, M., Kelly, G., & Saunders, R. (1999). Use and impact of satellite atmospheric motion winds on ECMWF analyses and forecasts. *Monthly Weather Review*, 127, 971–986.

ver Hoef, J. M., Cressie, N., & Barry, R. P. (2004). Flexible spatial models for kriging and cokriging using moving averages and the fast Fourier transform (FFT). *Journal of Computational and Graphical Statistics*, 13, 265–282.

WeatherOnline. (2018). *Calima*. WeatherOnline Ltd.-Meteorological Services. [https://www.weatheronline.co.uk/reports/wind/The-Calima.htm](https://www.weatheronline.co.uk/reports/wind/The-Calima.htm).

Wiens, A., Nychka, D., & Kleiber, W. (2020). Modeling spatial data using local likelihood estimation and a Matérn to spatial autoregressive translation. *Environmetrics*, 31, e2652.

How to cite this article: Sahoo, I., Guinness, J., & Reich, B. J. (2023). Estimating atmospheric motion winds from satellite image data using space-time drift models. *Environmetrics*, 34(8), e2818. [https://doi.org/10.1002/env.2818](https://doi.org/10.1002/env.2818)

**APPENDIX**

A. Standardization of data: We standardize the brightness temperature data $Y(x, t)$ as

$$Z(x, t) = \frac{Y(x, t) - \hat{\mu}(x)}{\hat{\sigma}(x)}$$

where $\hat{\mu}(x)$ denotes the pixel-wise sample mean over time of $Y(x, t)$ at location $x$, and $\hat{\sigma}(x)$ denotes the corresponding smoothed standard deviation. The standard deviation is smoothed using the Gaussian kernel

$$\phi(x|v_l, \lambda) = \frac{1}{2\pi\lambda^2} \exp\left(-\frac{||x - v_l||^2}{2\lambda^2}\right),$$
where \( \lambda > 0 \) is the kernel bandwidth and \( \mathbf{v}_1, \ldots, \mathbf{v}_L \in \mathbb{R}^2 \) denote the spatial locations over the entire region considered. The smoothing weights are defined as

\[
w_j(x) = \frac{\phi(x|v_j, \lambda)}{\sum_{j=1}^L \phi(x|v_j, \lambda)}.
\]

That is, for standardizing the data we use

\[
\tilde{\sigma}(x) = \sum_{i=1}^L \tilde{\sigma}(v_i)w_i(x),
\]

where \( \tilde{\sigma}(v_i) \) is the sample standard deviation at location \( v_i \).

**B. Smoothing the estimates:** Let \( \hat{\mathbf{u}}(x, t) = (\hat{u}(x, t), \hat{v}(x, t)) \) denote the estimated wind vector in \( D(x, t) \) using the proposed method. Also let the corresponding estimated variances be denoted by \( \hat{\sigma}_u(x, t) \) and \( \hat{\sigma}_v(x, t) \) where the suffixes ‘u’ and ‘v’ refer to the u- and v- component of the estimated wind vector. We smooth each component of the estimates using a weighted Gaussian smoothing filter, the weights being equal to the inverses of the corresponding variance estimates. Let \( \tilde{\mathbf{u}}^{(s)}(x, t) = (\tilde{\hat{u}}^{(s)}(x, t), \tilde{\hat{v}}^{(s)}(x, t)) \) denote the smoothed wind estimates where,

\[
\tilde{\hat{u}}^{(s)}(x, t) = \sum_{i=1}^L \hat{u}(v_i, t)w_i^{(s)}(x, t), \quad w_i^{(s)}(x, t) = \frac{\{\phi(x|v_i, \lambda)/\hat{\sigma}_u(v_i, t)\}}{\sum_{j=1}^L \{\phi(x|v_j, \lambda)/\hat{\sigma}_u(v_j, t)\}}
\]

\[
\tilde{\hat{v}}^{(s)}(x, t) = \sum_{i=1}^L \hat{v}(v_i, t)w_i^{(s)}(x, t), \quad w_i^{(s)}(x, t) = \frac{\{\phi(x|v_i, \lambda)/\hat{\sigma}_v(v_i, t)\}}{\sum_{j=1}^L \{\phi(x|v_j, \lambda)/\hat{\sigma}_v(v_j, t)\}}
\]

**C. Choosing appropriate window size:** We conducted additional simulation studies by examining a simple nonconstant wind field to demonstrate that choosing an arbitrarily large target window might lead to inaccurate estimates of the wind vectors. We consider a scenario where the winds diverge from the central meridian of a unit square spatial domain, that is, winds blow west in the west and east in the east. We assume that the wind speed increases quadratically from the center toward the respective edges. This simulation setting emulates divergent atmospheric winds, which are more prevalent in the upper atmosphere. Mathematically, we have,

\[
u_1(x, t) = \text{sign}(x_1 - 0.5) \times (x_1 - 0.5)^2, \quad u_2(x, t) = 0.
\]

Based on this wind pattern, data sets are generated on a \( [0, 1]^2 \) spatio-temporal grid of size \( 100 \times 100 \times 3 \) using the aforementioned fast Gaussian approximation algorithm with an exponential space-time covariance function \( f(t) \) with \( \nu = 1/2, \sigma_0^2 = 0.075 \) (4 pixels) and \( \sigma_2^2 = 5 \). We fit the proposed space-time drift model locally in space and estimate all parameters using multiple window sizes. Since we assume that the wind field does not vary along the vertical axis, the estimation is done along the horizontal axis at a subset of locations along the vertical axis.

Table A1 compares the average performance of the two methods over \( N = 50 \) simulated data sets. From the results, it can be seen that while STDM performs better than the DMW Algorithm for all window sizes, the performance of STDM deteriorates for windows of sizes larger than \( 25 \times 25 \). The estimated wind fields from both methods are shown in Figures 2 and 3.

| Method  | 7 \times 7 | 11 \times 11 | 15 \times 15 | 21 \times 21 | 25 \times 25 | 31 \times 31 | 35 \times 35 |
|---------|------------|-------------|-------------|-------------|-------------|-------------|-------------|
| STDM    | 3.128 (3.588) | 1.537 (2.271) | 0.905 (1.413) | 0.661 (1.153) | 0.619 (1.093) | 0.707 (1.139) | 0.818 (1.280) |
| DMWA    | 4.140 (1.681) | 3.414 (2.011) | 2.926 (2.180) | 1.935 (2.007) | 1.214 (1.517) | 0.998 (1.446) | 0.830 (1.320) |
**Figure A1** True (red) and estimated (blue) wind fields using the Space-time Drift Model (top row) and the DMW Algorithm (bottom row) for one representative data set, simulated using nonconstant wind field. The left column shows wind estimates calculated using window of size $25 \times 25$ and the right column shows the same calculated using window of size $35 \times 35$. The wind vectors are plotted for a subset of spatial locations for clear visualization.

In Figure A1, it can be seen that the estimates of the smaller wind vectors on both sides of the central meridian become less accurate with increasing window size. This accounts for the increase of the MVD and the associated SD for window sizes larger than $25 \times 25$. 