Hindawi
Advances in Meteorology
Volume 2020, Article ID 8841913, 12 pages
https://doi.org/10.1155/2020/8841913

Research Article

Quantifying the Location Error of Precipitation Nowcasts

Arthur Costa Tomaz de Souza, Georgy Ayzel, and Maik Heistermann

University of Potsdam, Institute of Environmental Science and Geography, Potsdam 14476, Germany

Correspondence should be addressed to Arthur Costa Tomaz de Souza; costatomazde@uni-potsdam.de

Received 11 September 2020; Accepted 21 October 2020; Published 3 December 2020

Academic Editor: Francesco Viola

In precipitation nowcasting, it is common to track the motion of precipitation in a sequence of weather radar images and to extrapolate this motion into the future. The total error of such a prediction consists of an error in the predicted location of a precipitation feature and an error in the change of precipitation intensity over lead time. So far, verification measures did not allow isolating the extent of location errors, making it difficult to specifically improve nowcast models with regard to location prediction. In this paper, we introduce a framework to directly quantify the location error. To that end, we detect and track scale-invariant precipitation features (corners) in radar images. We then consider these observed tracks as the true reference in order to evaluate the performance (or, inversely, the error) of any model that aims to predict the future location of a precipitation feature. Hence, the location error of a forecast at any lead time Δt ahead of the forecast time t corresponds to the Euclidean distance between the observed and the predicted feature locations at $t + \Delta t$. Based on this framework, we carried out a benchmarking case study using one year worth of weather radar composites of the German Weather Service. We evaluated the performance of four extrapolation models, two of which are based on the linear extrapolation of corner motion from $t - 1$ to t (LK-Lin1) and $t - 4$ to t (LK-Lin4) and the other two are based on the Dense Inverse Search (DIS) method: motion vectors obtained from DIS are used to predict feature locations by linear (DIS-Lin1) and Semi-Lagrangian extrapolation (DIS-Rot1). Of those four models, DIS-Lin1 and LK-Lin4 turned out to be the most skillful with regard to the prediction of feature location, while we also found that the model skill dramatically depends on the sinuosity of the observed tracks. The dataset of 376,125 detected feature tracks in 2016 is openly available to foster the improvement of location prediction in extrapolation-based nowcasting models.

1. Introduction

Forecasting precipitation for the imminent future (i.e., minutes to hours) is typically referred to as *precipitation nowcasting*. A common nowcasting technique is to track the motion of precipitation from a sequence of weather radar images and to extrapolate that motion into the future [1]. For that purpose, we often assume that the intensity of precipitation features in the most recent image remains constant over the lead time period—an assumption commonly referred to as "Lagrangian persistence" [2]. In Lagrangian *field tracking*, a velocity vector is obtained for each pixel of a precipitation field, and that vector field is used to extrapolate the motion of the entire precipitation field—as opposed to *cell tracking* in which contiguous high-intensity objects are tracked (see [3] for a discussion of both methods).

The present study focuses on nowcasts that are based on *field tracking*. The performance (or skill) of field tracking techniques is mostly verified by comparing the forecast precipitation field $F_{t+\Delta t}$ for time $t + \Delta t$ against the observed precipitation field $O_{t+\Delta t}$ at time $t + \Delta t$, where t is the forecast time and Δt is the lead time. A large variety of verification measures have been suggested in the literature (see, e.g., [4, 5]). Most of them, however, struggle with disentangling different sources of error: when we compare $F_{t+\Delta t}$ to $O_{t+\Delta t}$, how can we know the cause of the disagreement? Was it our prediction of the future location of a precipitation feature, or was it how precipitation intensity changed over time? Some verification scores, such as the Fractions Skill Score [6], apply a metric over spatial windows of increasing size in order to examine how the forecast performance depends on the spatial scale. Yet, we still lack the ability to explicitly

isolate and quantify the location error. This makes it difficult to benchmark and optimize the corresponding components of nowcast models.

In this study, we introduce an approach to directly quantify the location error of precipitation nowcasts which is based on the extrapolation of field motion. With *location error*, we refer to the spatial offset (or Euclidean distance) between the true and the forecast locations of a precipitation feature (Figure 1). In this context, the term "feature" does not refer to a contiguous object but to a distinct *point* in the precipitation field, and we make use of the ability of the OpenCV library to detect and track the true motion of such distinct points. In a verification case study, we will demonstrate the ability to quantify the location error by benchmarking a set of routine extrapolation techniques for one year of quality-checked radar data in Germany.

Section 2 highlights the approach to quantify the location error and describes a set of tracking and extrapolation techniques based on optical flow, as well as the radar data for our case study. Section 3 presents the results of our case study, and Section 4 concludes the paper.

2. Methods and Data

2.1. Feature Detection and Tracking. We suggest quantifying the location error of a forecast by comparing the observed location (or displacement) of a precipitation feature against its predicted location. In visual computing, a feature is defined as a point that stands out in a local neighborhood and is invariant in terms of scale, rotation, and brightness [7]. For a radar image, a feature (or corner) represents a point with a sharp gradient of rainfall intensity [2].

In this study, features are detected using the approach of Shi and Tomasi [8] If a feature is detected at one time step, we attempt to track that feature in any subsequent time step until it is no longer trackable. The feature tracking follows the approach of Kanade [9], as implemented by Bouguet [10]. The tracking *error* (or, inversely put, the robustness of tracking a feature from one radar image to the next) is quantified in terms of the minimum eigenvalue of a 2×2 normal matrix of optical flow equations (this matrix is called a spatial gradient matrix in Bouguet [10]), divided by the number of pixels in a neighborhood window. In the tracking step, that minimum eigenvalue has to exceed a threshold in order for a feature to be considered as successfully tracked. Table 1 provides an overview of parameters used for both feature detection *and* tracking. These values are based on the ones presented by Ayzel et al. [2]. The underlying equations are well documented in OpenCV [11].

In order to increase the robustness of track detection, the tracking was also performed backwards at each time step (Figure 2): let p_t signify a feature that was identified at frame t and tracked to the next frame at time $t+1$ at the position p_{t+1}. The same tracking process was then applied backwards from the point p_{t+1} to time t, yielding the point p_t^{back}. Only the trajectories where the distance d_{back} between the source point p_t and the backwards tracked point p_K^{back} was less than one kilometer (the grid resolution) were considered in our analysis.

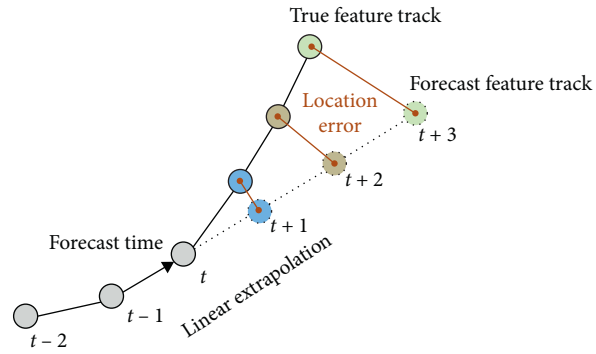

Figure 1: Illustration of the location error for a prediction at forecast time t that is based on the linear extrapolation of feature motion from $t-1$ to t.

To collect all feature tracks T in any given time period with a length of n time steps, we detect "*goodFeaturesToTrack*" (Shi and Tomasi, 1994) at each time step $k \in [1, \ldots, n]$, and track these features over as many subsequent time steps as possible. Accordingly, each track $T_{i,j,k}$ could be identified by a unique tuple (i, j, k) that carries its starting point (by the grid's row and column indices, i and j) and its starting time index k. In this study, we use an analysis period of one year (2016, a leap year) and a time step length of 5 minutes, so that $k \in [1, \ldots, 105408]$.

In summary, the tracking process consists of six steps:

(1) Identify the features p_k^{new} using *goodFeaturesToTrack* [11] at any time k.

(2) If there are already features being tracked, p_k^{old}, from $k-1$ to k, we consider only those features p_k^{new} for which the distance to any feature p_k^{old} is greater than 7 km (with this threshold, we enforce consistency with the minDistance parameter of the Shi-Tomasi corner detection; see Table 1). The trackable features p_k are hence the union of p_k^{old} and p_k^{new}.

(3) Track p_k from k to $k+1$ using *calcOpticalFlowPyrLK* [11];

(4) Backwards track (from $k+1$ to k) those features p_{k+1} that were obtained in step 3.

(5) Calculate the distance, d_{back}, from the features p_k to the backward-tracked locations resulting from step 4.

(6) Keep only those features p_{k+1} where the distance d_{back} is less than 1 km. These features are now p_{k+1}^{old}.

For statistical analysis, each track $T_{i,j,k}$ is characterized by its duration τ (the number of time steps over which the track persists), the overall displacement distance d of the feature along its track, the average feature velocity $v = d/\tau$, and the straightness of the feature's displacement in terms of the sinuosity index (SI) (which is calculated by dividing d by the Euclidean distance between the feature origin and end locations). The concept of sinuosity is widely used to characterize river curvatures as introduced by Mueller [12] and was also applied to atmospheric science by Terry and Feng

Table 1: OpenCV function parameters used for feature detection and tracking.

Parameter name	Value	Meaning
maxCorners	200	Maximum number of features
qualityLevel	0.2	Minimum accepted quality of features
minDistance	7	Minimal Euclidean distance between features
blockSize	21	Size of pixel neighborhood for covariance calculation
winSize	(20, 20)	Size of the search window
maxLevel	2	Maximal number of pyramid levels

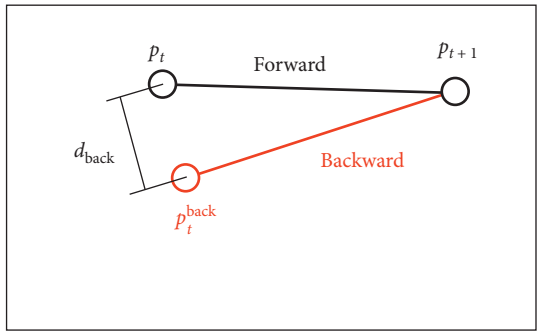

Figure 2: Illustration of the backward tracking test performed at each time step for all features.

Table 2: Overview of extrapolation models.

Name	Main approach	# Time steps looking back
Persist	Eulerian persistence	0
LK-Lin1	Linear extrapolation based on Lukas Kanade	1
LK-Lin4	Linear extrapolation based on Lukas Kanade	4
DIS-Lin1	Linear extrapolation from DIS motion field	1
DIS-Rot1	Semi-Lagrangian extrapolation based on motion field obtained by dense optical flow	1

[13] to quantify the sinuosity of typhoon tracks. In our analysis, we will also use the sinuosity index in order to understand the error of predicted feature locations.

2.2. Error of Predicted Locations. Let p be the true location and let P be the predicted location of a point feature in a Cartesian coordinate system. At forecast time t, p_t will be equal to P_t. Consider $P_{t+\Delta t} = f(p_t, \Delta t, S_t)$ any function or algorithm that predicts the future location $P_{t+\Delta t}$ of point p_t from any set S_t of predictors that is available at time t or before. In the context of our study, that set of predictors could be, for example, the previous locations $p_{t-1}, p_{t-2}, \ldots$ of p_t. We then define the error of our prediction, henceforth referred to as *location error* ε, as the Euclidean distance between $P_{t+\Delta t}$ and $p_{t+\Delta t}$.

2.3. Extrapolation Techniques. In a verification experiment, we can use our collection of tracks T in order to retrieve points p_t for which the location $P_{t+\Delta t}$ at $t+\Delta t$ should be predicted, points that could be used as predictors (S_t), as well as the true location $p_{t+\Delta t}$ of the point at $t+\Delta t$. Assuming that an extrapolation of motion uses feature locations from m time steps before t, the minimum feature track length to produce a forecast would be $m+1$. In order to retrieve the location error of such a prediction at time $t+\Delta t$, we would need a minimum track length of $m+\Delta t+1$.

Based on the above terminology, we present in the following the extrapolation models analyzed in the present study. These models are based on the models that were also evaluated in a recent benchmarking study on optical-flow-based precipitation nowcasting [2]. Table 2 gives an overview of model acronyms and their main properties.

2.3.1. Eulerian Persistence. As a trivial benchmark, we use the assumption of Eulerian persistence, meaning that the precipitation feature will simply remain at its position at forecast time; that is, $P_{t+\Delta t} = p_t$.

2.3.2. Linear Extrapolation. Linear extrapolation of feature motion assumes that a feature moves, over any lead time, at constant velocity and in the same direction. The displacement vector representing this motion can be obtained in different ways. These ways constitute three different models exemplified in the present study: *LK-Lin1*, *LK-Lin4*, and *DIS-Lin1*. In the case of *LK-Lin1* and *LK-Lin4*, the displacement vector is obtained from "looking back" m time steps from forecast time t to previous feature locations at $t-m$ (tracked by using the Lucas–Kanade method, hence the LK label). For LK-Lin1, m equals 1, so the vector $v(t, p_t)$ to displace feature p_t is the connection from p_{t-1} to p_t; for LK-Lin4, m equals 4, so that the displacement vector results from the connection between p_{t-4} and p_t, where the length of the vector is divided by 4 in order to obtain the displacement velocity. Hence, a forecast at lead time Δt extends the vector $v(t, p_t)$ correspondingly. Please see Figure 3 for an illustration of both the LK-Lin1 and the LK-Lin4 method. Of course, any other look-back time m could be used to obtain a displacement vector. In this study, we arbitrarily used $m \in \{1, 4\}$ in order to examine the effect of m on the forecast performance.

For the DIS-Lin1 model, a complete field of motion vectors $\mathbf{V}_{\text{DIS}}$ is obtained from the Dense Inverse Search (DIS) method [14]; the underlying concept and equations of the DIS method have been elaborated by Kroeger et al. [15] and then used for the extrapolation. A point p_t is linearly extrapolated from t to $t+n$ by n times the velocity vector $v_{\text{DIS}}(t, p_t)$, where $v_{\text{DIS}}(t, p_t)$ is the vector closest to p_t in the

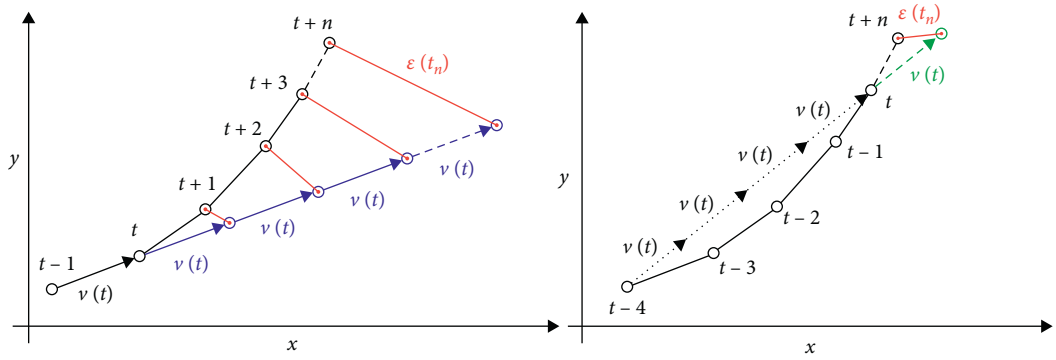

FIGURE 3: Illustration of the linear extrapolation schemes for the LK group: on the left LK-Lin1 and on the right LK-Lin4. The location error is displayed by $\varepsilon(t_n)$.

$\mathbf{V}_{\mathrm{DIS}}(t)$ field (Figure 4). $\mathbf{V}_{\mathrm{DIS}}(t)$ is calculated by OpenCV's *cv2.DISOpticalFlow_create* function, which returns velocity vectors for each grid pixel based on the radar frames from $t-1$ to t. In a recent benchmarking study about optical-flow-based precipitation nowcasting, Ayzel et al. [2] showed that the DIS-based model (referred to as the "Dense" model in that paper) is an effective method for radar-based precipitation nowcasting.

2.3.3. Semi-Lagrangian Approach Based on Dense Optical Flow. In a Semi-Lagrangian approach, the motion field is typically assumed as constant over the forecast period and the feature trajectory is determined by following the streamlines [16]. Following this concept, the DIS-Rot1 model (corresponding to "Dense rotation" in [2]) uses the two most recent radar images, $t-1$ and t, to estimate $\mathbf{V}_{\mathrm{DIS}}(t)$ by cv2.DISOpticalFlow_create function. Similar to the DIS-Lin1 model, the displacement vector $v_{\mathrm{DIS}}(t, p_t)$ which is closest to p_t is used to extrapolate the motion of p_t from its position at t to $t+1$, providing the location of P_{t+1}. This process is repeated at all lead time steps until the maximum lead time is achieved. Hence, at each lead time step n, we retrieve the vector $v_{DIS}(t, P_{t+n})$ which is closest to P_{t+n} in order to extrapolate the feature location, P_{t+n+1}. Accordingly, the velocity vector is updated at each lead time step from $\mathbf{V}_{\mathrm{DIS}}(t)$, allowing for rotational or curved motion patterns (Figure 5).

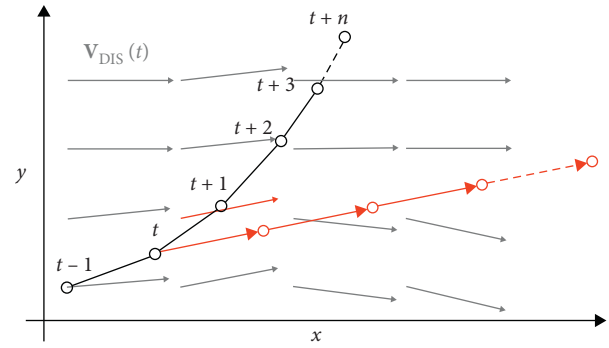

FIGURE 4: In the DIS-Lin1 model, the vector $v_{\mathrm{DIS}}(t, p_t)$ (light red arrow) obtained from $\mathbf{V}_{\mathrm{DIS}}(t)$ is transferred to the p_t location and linearly extended to $t+n$.

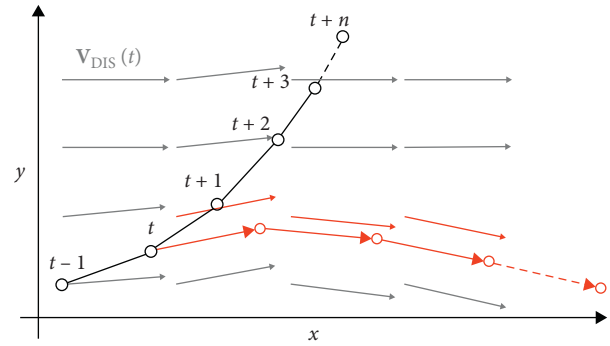

FIGURE 5: Schematic of the DIS-Rot1 model (orange path), where the velocity is updated every time step by transferring the velocity vector $v_{\mathrm{DIS}}(t, p)$ (light orange arrow) closest to p_t (black circles, for $t=0$) or P_{t+1} (orange circles, for $t>0$) in $\mathbf{V}_{\mathrm{DIS}(t)}$, to the $P_t + \Delta t$ location to advect.

2.4. Weather Radar Data and Experimental Setup. Our benchmarking case study is based on weather radar data from the German Weather Service, namely, the RY product generated as part of the RADKLIM radar reanalysis of the German Weather Service DWD [17]. The RY product represents a quality-controlled national precipitation intensity composite from 18 C-Band radars covering Germany at 5-minute intervals and a spatial resolution of 1 km at an extent of 1100×900 km. The basis of the composite product is the so-called "precipitation scans" from each of the 18 radar locations. The precipitation scan is designed to follow the horizon as closely as possible at an azimuth resolution of 1° and a radial resolution of 1 km, adjusting the elevation angle for each azimuth depending on the presence of mountains that would interfere with the beam propagation. Quality control includes a wide range of correction methods for, e.g., clutter or partial beam blockage (see [17] for details).

The year 2016, selected for this experiment, was characterized by an annual precipitation close to the climatological mean for most regions in Germany, as can be seen in the German Climate Atlas [18]. However, the precipitation

mean during autumn was below the normal average and during the winter months slightly above the climatological mean.

As 2016 was a leap year, this experiment was carried out on 105408 radar composite images. Since none of the methods under evaluation required any kind of training, there was no need to split the data into sets for calibration and validation. Instead, we used all tracks for verification. For each track, we always use, as forecast time t, a time of 20 minutes after the feature was detected for the first time. That is because our model LK-Lin4 needs to look back four time steps (i.e., 20 minutes) in order to make a forecast, and we need to make sure, for a fair comparison, to compare all models for the same forecast times.

2.5. Computational Details. The analysis was carried out in a Python 3.6 environment using the following main open-source libraries: NumPy (https://numpy.org), NumExpr (https://github.com/pydata/numexpr), and SciPy (https://www.scipy.org) for general computations; OpenCV (https://opencv.org) for feature tracking; and Pandas (https://pandas.pydata.org) and h5py (https://www.h5py.org).

3. Results and Discussion

3.1. Properties of Collected Tracks. The identification and tracking process detected 376,125 features above the rainfall rate threshold of 0.2 mm/h and lasted over 20 minutes, which resulted in 337,776 eligible tracks after applying the extrapolation step. A track was considered as "eligible" in case all models had a predicted location at all lead times, from t to $t + n$. The loss of 10.2% that is implied by the above numbers was caused by the DIS group of models which did not generate a valid velocity vector $v_{\mathrm{DIS}}(t, p_t)$ near every p_t point, in the $\mathbf{V}_{\mathrm{DIS}}(t)$ field, within a 3.5 km threshold.

Figure 6 gives an overview of the properties of the valid tracks. The figure also shows the seasonal dependency of these track properties by summarizing their distribution on a per month basis. We would like to emphasize that this analysis must not be interpreted as a "climatology" of track properties as it only contains data from a single year. Still, we consider it as illustrative to investigate which properties tend to exhibit a seasonal pattern and also to discuss whether the observed properties can be considered as representative for the governing rainfall processes in Germany.

In an average month of 2016, we identified and tracked 28,146 features (Figure 6(a)). The largest number of tracks is found from April to August (all above the average). Yet, there is no continuous seasonal pattern in the number of detected tracks because, e.g., January and October also show rather large counts.

No pattern at all can be found for the track length (Figure 6(b)). With an average track length of 128 km, monthly maximum mean and median track lengths occur in January, April, and September. A partly similar pattern can be found for the track duration that amounts to 207 minutes on average (Figure 6(c)). This is plausible as we would, in general, expect the length of a track to increase with its duration. Yet, there are also months—most notably the summer months from May to August—where this expectation is not met; and, of course, the length of a track depends not only on its duration but also on a feature's velocity. The average feature velocity in 2016 amounted to a value of 42 km/h; and, in fact, not only does velocity show a clear seasonal pattern (with minimum velocities in the summer months; see Figure 6(d)), but also the seasonal pattern helps us to understand where the patterns of track length and duration appear to be "inconsistent." For example, the track velocity is at a minimum in May and June, which decreases the length of track despite the rather high duration values for these two months.

The clearest seasonal pattern can be observed for rainfall intensity (Figure 6(e)). That pattern is very much in line with our expectation as rainfall in the summer months is governed by convective events that tend to be more intense than stratiform event types. However, if we assumed that a higher rainfall intensity along a track is caused by the convective nature of the underlying event, the track duration in the corresponding months (e.g., May and June) is at least surprising: we would expect a convective event not only to be more intense but also to be rather short (in comparison to widespread stratiform rainfall). The apparent inconsistency between the patterns of rainfall intensity and track duration points us to one of the key issues with the presented track inventory: we must not misinterpret a "track" as an "event" in a hydrometeorological sense. The corner detection algorithm (see Section 2.1) searches for pronounced features in the sense of strong local gradients and tracks a feature for as long as it stands out. While we define a rainfall event as some coherent process in space and time, the tracking algorithm could "lose" a feature right in the course of an ongoing event and maybe, at the same time, find another feature to track somewhere else in the field. Obviously, the tracking algorithm was able to track features over a longer duration in May, June, and September of 2016. However, as of now, we do not know which properties of the corresponding rainfall events caused that effect. We should just emphasize that the duration of a track does not necessarily correspond to the duration of an event. In the same way, we cannot expect the tracking algorithm to find features at "representative" locations of a convective cell. It will detect such features anywhere in a rainfall field where local gradients meet the tracking criteria. That could be right not only in the middle of heavy rainfall but also at the edges. Hence, the reported precipitation intensities along the tracks will not be representative of the mean precipitation intensities of the corresponding precipitation fields.

Altogether, we have to emphasize at this point that the seasonal track statistics are indeed plausible. But it must be clear that track statistics are not necessarily representative for "event" statistics. That notion might be irritating for those who have been defining and tracking features in terms of coherent rainfall objects over their lifetime from initiation to dissipation. A new feature track as we understand it in our analysis could be found right in the middle of an ongoing event, and it can be lost long before the actual rainfall

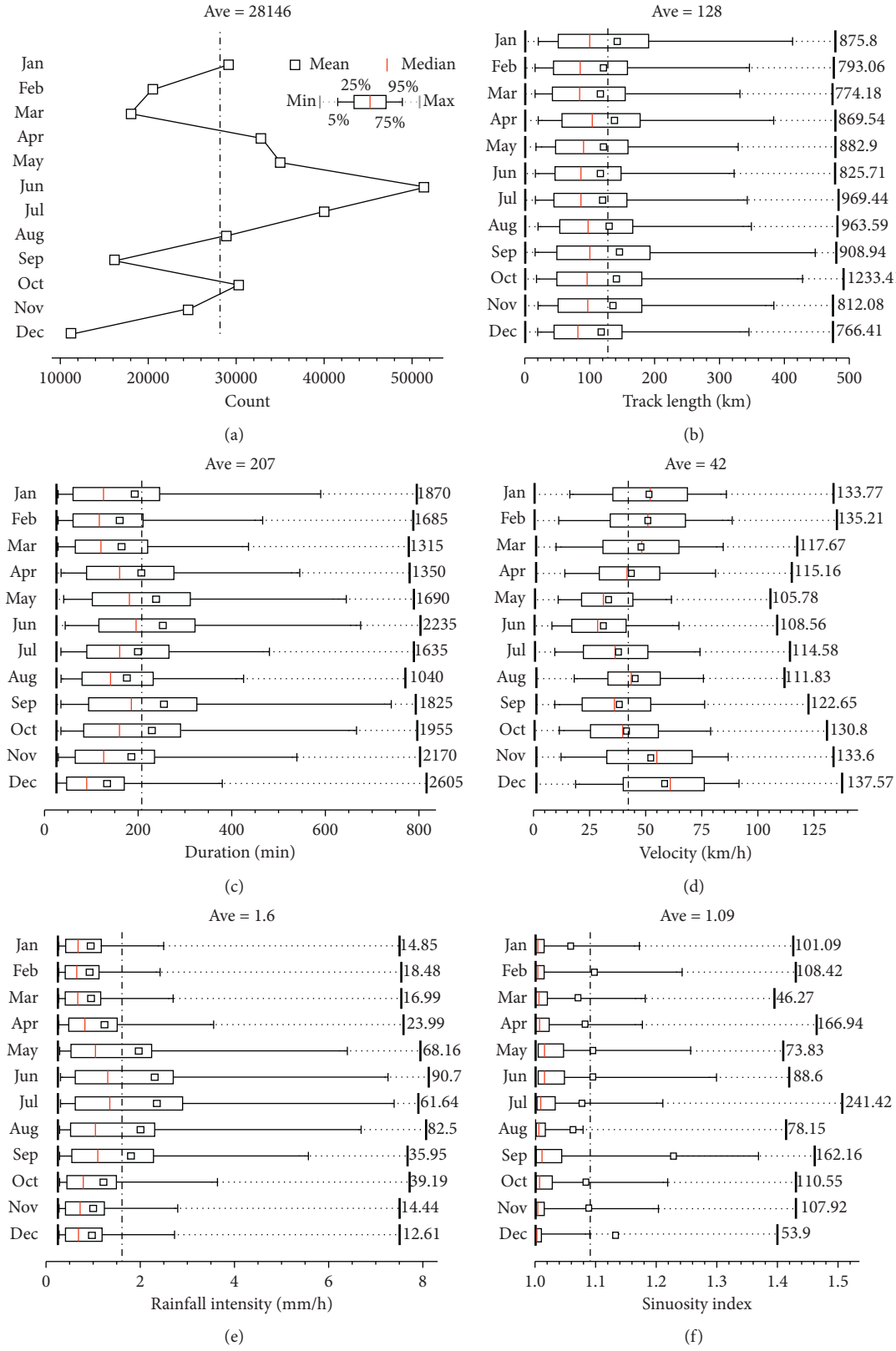

Figure 6: Statistical properties of detected tracks, organized by month: (a) number of detected tracks, (b) track length, (c) track duration (time elapsed from detection and loss of a feature), (d) feature velocity, (e) rainfall intensity of a detected feature, and (f) sinuosity index of a track.

"object" dissolves. However, that does not at all lessen the value of these tracks for the purpose of our analysis, which is to quantify the forecast location error based on well-defined and scale-invariant features.

Having said that, one final track property shown in Figure 6(f) has not been discussed yet: the sinuosity index. As pointed out above (Section 2.1), the sinuosity index illustrates how much the shape of a track deviates from a straight line (which would correspond to a sinuosity index of 1). Figure 6(f) shows rather large sinuosity values for the summer months, May to September, but there is no obvious seasonal pattern. More strikingly, the distribution of the sinuosity index is very heavily tailed. The average value amounts to approximately 1.10 in the year 2016, which is, at the same time, the 90th percentile of the sinuosity values. That means, in turn, that the vast majority of tracks are rather straight, while the remaining tracks show all kinds of curved, meandering, twisted, or just erratic behavior.

Hence, before we systematically show the results of our verification experiment with regard to the location error (see Section 3.3), we would like to illustrate, in the following paragraph, the behavior of observed tracks in comparison to the forecast tracks under different sinuosity conditions.

3.2. Visual Examples of Observed and Predicted Tracks. Before we systematically evaluate the performance of different extrapolation techniques, we would like to provide some illustrative examples of observed versus predicted tracks. The selection of tracks for this illustration is arbitrary and does not intend to be representative of the performance of any of the extrapolation methods. Instead, we aim to exemplify shapes of observed and predicted tracks under different sinuosity conditions in order to convey a better understanding of the various constellations that will finally be condensed into one single location error value.

Figure 7 shows a "gallery" of 11 observed tracks in different subplots (From Figure 7(a) to 7(k)). Each subplot also contains the tracks that were predicted by the different extrapolation models. Each dot represents one feature location in a 30-minute time step, except the first one that represents the first prediction step at five-minute lead time. LK-Lin1 and LK-Lin4 infer the displacement vector directly from the feature positions at t and $t-1$ or t and $t-4$, respectively. As a reminder, DIS-Lin1 and DIS-Rot1 obtain the displacement vector of a feature from the DIS algorithm, a dense optical flow technique that produces motion fields based on the radar images at t and $t-1$; DIS-Lin1 extrapolates the closest vector linearly over the entire lead time, while DIS-Rot1 uses a Semi-Lagrangian scheme in which the displacement vector is updated as the feature moves through the velocity field obtained from the DIS technique. Further details have been provided in Section 2.3. As in all forecasts of our verification experiment, the forecast time t corresponds to the 5th feature of the observed track. That is because the LK-Lin4 method needs to look four steps back in time ($t-4$) in order to produce a forecast, while the other methods only look back one step in time ($t-1$).

In order to convey a better idea about the rainfall patterns in the examples, the observed rainfall intensity at forecast time t is plotted as a background in grey scale. Furthermore, the sinuosity index and the track duration are printed in the corresponding subplots.

Please note that the duration of the observed tracks in Figure 7 can extend over many hours; very long tracks were capped at a duration of 300 minutes for the purpose of plotting. Furthermore, the lead time of the predictions in the examples was set to the (capped) track duration minus 20 minutes (which corresponds to the period $t-4$ until forest time t). As a consequence, the lead times illustrated in Figure 7 are mostly longer than the maximum lead time of 120 minutes, which is used in our verification experiment (see the next section). Hence, the first visual impression of Figure 7 is dominated by the considerable errors that can occur for such long lead times. But, of course, we should rather be aware of the behavior for shorter lead times up to 120 minutes. For that reason, the 120-minute lead time is highlighted by a larger dot.

Not surprisingly, most of the competing methods appear to remain rather close to the observed track for short lead times of up to 30 minutes (except, e.g., in subplot Figure 7(j) in which the DIS-based methods entirely fail to capture the direction of feature movement). After that, the lead time over which the extrapolation models adequately predict the observed feature track varies, depending on the persistence of the motion behavior and the validity of the underlying model assumption. For example, all models perform quite well for very long times in subplot (f). In subplot (i), the Semi-Lagrangian approach (DIS-Rot1) shows a clear advantage, while in subplots of Figures 7(c) and 7(k), DIS-Rot1 is outperformed by all other models. Surely, there are several examples (Figures 7(b), 7(d), 7(e), and 7(g)) in which all models entirely fail to anticipate the motion for lead times beyond 120 minutes.

As this compilation of examples is deliberately arbitrary, it does not provide a basis to infer the general superiority or inferiority of one or the other method. All models appear to struggle with predicting very sinuous tracks (subplots in Figures 7(b), 7(d), 7(e), and 7(g)), which is what we would expect. However, while the figure makes it difficult to compare the absolute location error between the examples (due to the different scales), it still appears that the absolute location error does not necessarily depend on the sinuosity. For example, the location error of LK-Lin1 after the maximum lead time (280 minutes) is higher in subplot 7(i) (almost straight, SI = 1.01) than it is in subplot 7(d) (SI = 1.36). In fact, straight tracks can imply a large error if the initial motion vector of a forecast method fails to represent the average long-term direction (see subplot 7(j) for a very impressive example). Then again, large errors can occur if a strong sinuosity of the track coincides with a large overestimation of the absolute velocity (e.g., subplots 7(b) and 7(g)). In that case, the linear extrapolation quickly departs from the track origin, while the actual feature track meanders slowly and remains in the close vicinity of the origin. For such a scenario, the trivial persistence model (the feature just remains at the origin) will be superior even for short lead times.

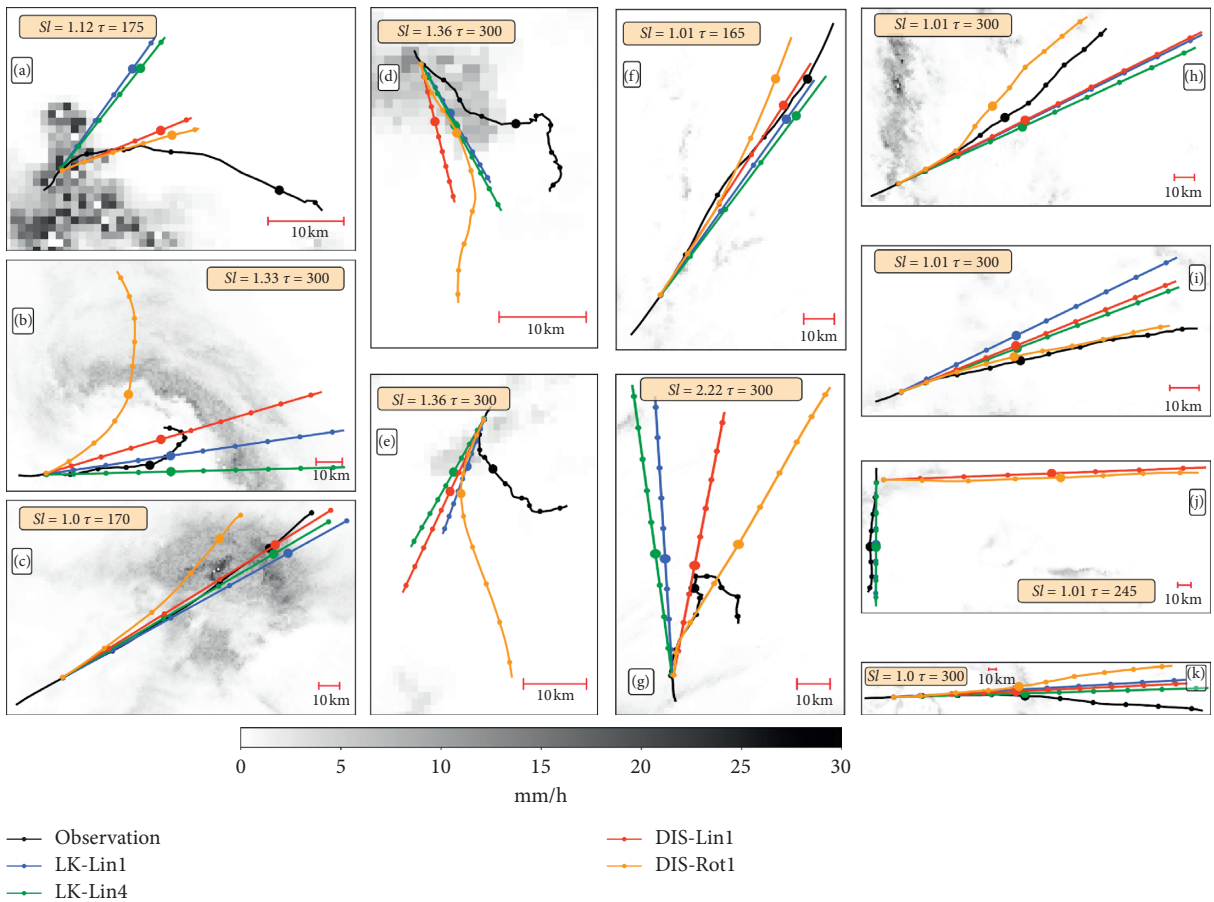

FIGURE 7: Compilation of forecast versus observed tracks under different sinuosity conditions. Due to the different spatial extents of the windows, the scale of each subplot is different. Hence, a 10 km scale bar is provided for orientation. For each example, the observed track duration τ (in hours) and its sinuosity index SI are shown. The lead time of 120 minutes is highlighted by a larger dot. Some very long tracks have been capped at a maximum of 300 minutes for illustrative purposes.

Altogether, these different examples give us a better idea of how location errors can develop from both inadequate model assumptions (e.g., linear approximation versus curved or sinuous conditions) and a failure to approximate the average motion from the initial feature locations. It is impossible, though, to diagnose the superiority of one or the other model from these examples. Hence, we will now systematically examine the results of our model verification experiment. We will not only analyze how the location error depends on lead time, but we will also investigate how the model performance relative to the persistence model depends on the sinuosity of the underlying tracks.

3.3. Systematic Quantification of the Location Error. After having exemplified different observed and predicted tracks in the previous section, we now present the results of our benchmarking experiment. Figure 8 shows the distribution of locations errors for different models and lead times up to 120 minutes. For each lead time, the box plots specify mean, median, interquartile range, and the 5th and 95th percentiles of the location error. For all models, the error quantiles increase slightly exponentially but almost linearly with lead time. The rate at which the location error grows with lead time is, for all models, dramatically lower than that for the persistence model; the mean error of persistence is higher than the mean error of any model at any lead time, which means that all models, *on average*, have positive skill at all lead times. For all models, the error distribution is obviously positively skewed, with the mean error being much higher than the median, and thus there is a heavy tail towards high location errors.

For very short lead times of up to 10 minutes, the mean error is about one kilometer for all competing models except for persistence which is already up at more than seven kilometers after ten minutes. After 60 minutes, the mean location error of all models exceeds a distance of 5 kilometers, as well as 10 kilometers after 110 minutes. For all models, at least 25% of all forecasts exceed an error of 5 kilometers after 50 minutes and an error of 10 kilometers after 90 minutes. After 75 minutes, at least 5% of all forecasts exceed an error of 15 kilometers.

Altogether, the location error can be substantial for a significant proportion of forecasts, while the median location error grows at a more moderate rate.

While this general pattern governs the behavior of all models, there are clear differences between the performances of the competing models. These differences, however, are not always coherent across all error quantiles and lead times,

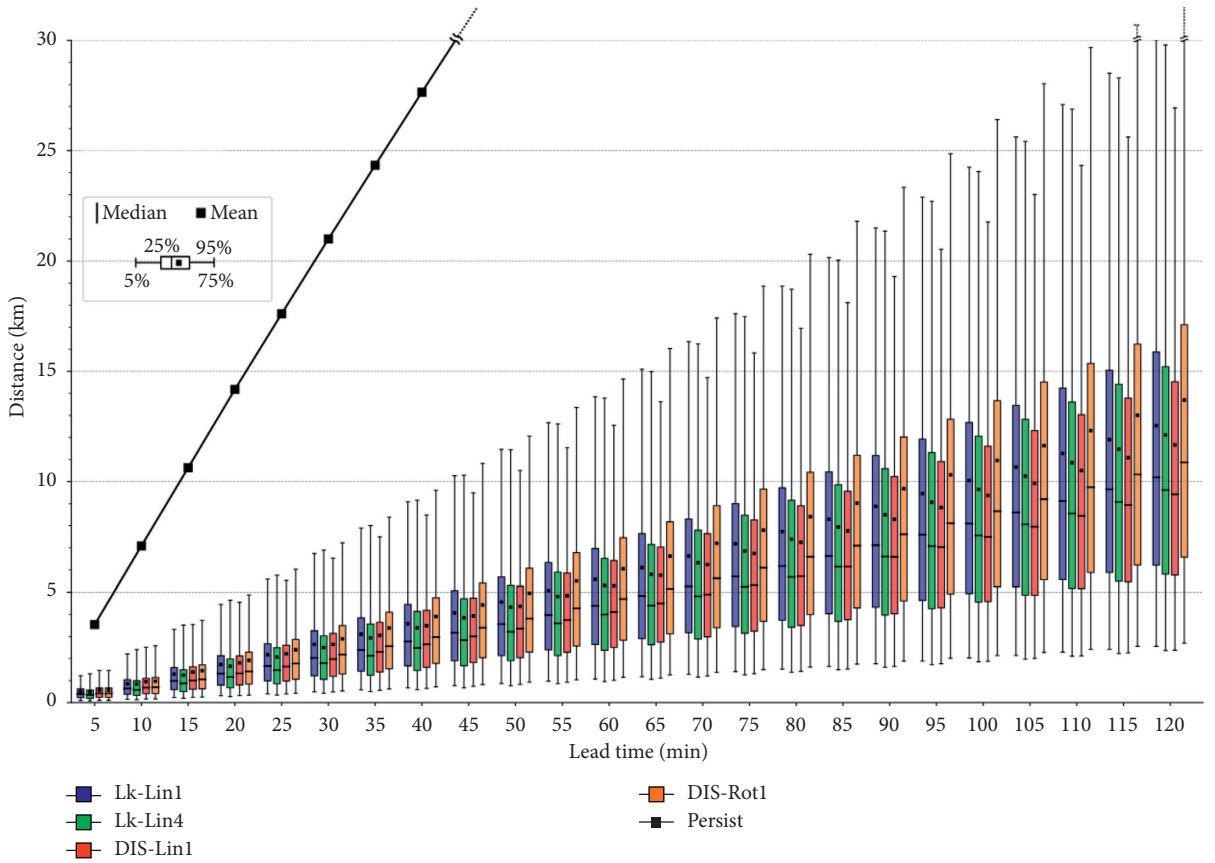

Figure 8: The distribution of location errors for different extrapolation models and lead times.

except for the DIS-Rot1 model, which has the weakest performance of all models at virtually all lead times and for all quantiles, and the LK-Lin1 model, which performs better than DIS-Rot1 but ranks second last. As for the best forecast performance, the LK-Lin4 and the DIS-Lin1 models take turns depending on error quantile and lead time: For the 5th and the 25th percentiles, the LK-Lin4 model performs best for lead times up to 100 minutes, for the median up to 80 minutes, and for the mean up to 55 minutes. The DIS-Lin1 model shows the strongest changes of relative performance over lead time: as for the mean error, DIS-Lin1 starts to outperform LK-Lin4 at a lead time of 60 minutes and continues this way until the maximum lead time of 120 minutes. As for the median error, DIS-Lin1 only catches up with LK-Lin4 after 90 minutes. For the 75th percentile, DIS-Lin1 outperforms LK-Lin4 after 50 minutes and for the 95th percentile already after 20 minutes. In summary, LK-Lin4 tends to outperform DIS-Lin1 in the first hour, while DIS-Lin1 becomes superior in the second hour, apparently because it tends to avoid very high errors more efficiently than LK-Lin4 does.

In the following, we would like to better understand how model skill is affected by sinuosity. In Section 3.2, we have already indicated that the absolute values of location errors do not clearly depend on sinuosity. That was confirmed by the systematic verification experiment (results not shown). Yet, the *difference* between an extrapolation model and the (trivial) persistence model might very well depend on sinuosity. In order to formally evaluate that hypothesis, we now examine the *skill* of our models more closely. Skill scores rate the score of a forecast in relation to the score of a reference forecast, in our case persistence. They are particularly useful in benchmark studies such as the present one. Equation (1) shows the general definition of skill as derived from any forecast score, as well as the specific formula if we use the location error ε as the "score" (which becomes zero for a perfect forecast) and persistence as the "reference":

$$\text{Skill} = \frac{\text{Score}_{\text{forecast}} - \text{Score}_{\text{reference}}}{\text{Score}_{\text{perfect}} - \text{Score}_{\text{reference}}} = \frac{\varepsilon_{\text{forecast}} - \varepsilon_{\text{persistence}}}{-\varepsilon_{\text{persistence}}}. \quad (1)$$

We examine the forecast skill under different sinuosity conditions. As already pointed out in Section 3.1, the distribution of sinuosity is highly skewed and 90% of observed tracks would pass as at least "rather straight" with a sinuosity index equal to or lower than 1.1. Hence, we split the forecasts into three unequal groups, depending on quantiles of the sinuosity index: The first group contains the "straight" 90% of the forecasts with a sinuosity index below 1.1. We consider the value of 1.1 as an—admittedly—arbitrary threshold between "rather straight" and "rather winding" tracks. The remaining 10% of tracks are split into two equally sized groups, again based on sinuosity: the 5% with the highest sinuosity, exceeding an SI value of 1.2, could be labelled as "twisted," and the remaining 5% with intermediate SI values between 1.1 and 1.2 could be labelled as "winding." Figure 9

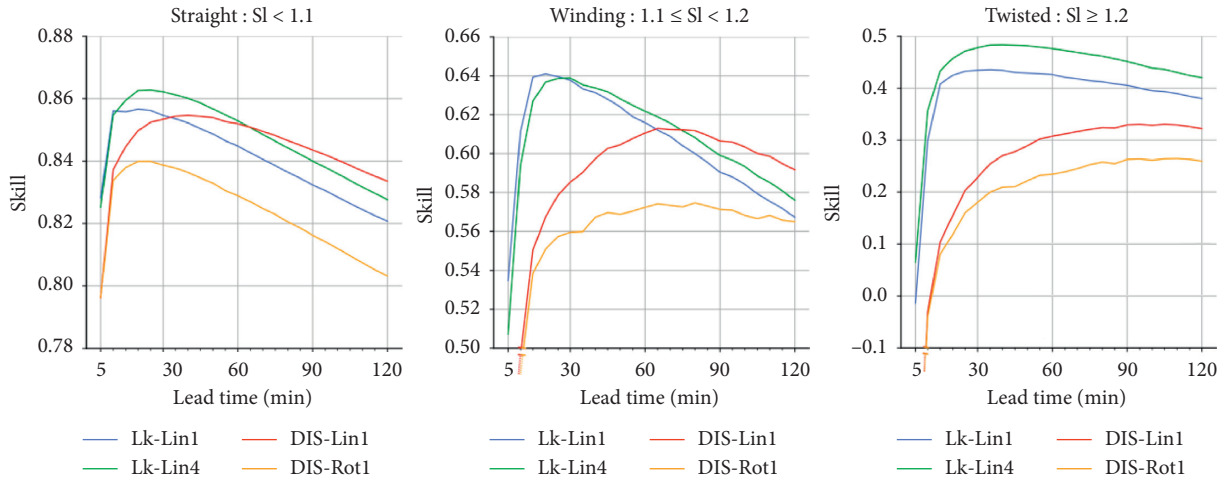

Figure 9: The mean model skill over each lead time with regard to location prediction for different extrapolation models and sinuosity conditions. Please note that the very low skill values of the DIS-based models at 5-minute lead time (in the winding and twisted groups) are hidden by the scaling of the y-axis. At five-minute lead time, both models only have a skill of about 0.35 (winding) and −0.55 (twisted).

shows the average model's skill over every lead time for these three sinuosity classes. Clearly, the model skill dramatically varies between these three groups: it ranges between 0.79 and 0.87 for the "straight" category, mostly between 0.5 and 0.65 for the "winding" category, and mostly between 0 and 0.5 for the "twisted" category. This decrease of skill with increasing sinuosity is well in line with our expectation. Furthermore, the ranking of all models based on skill is quite coherent across all categories and also consistent with our previous analysis of location errors. DIS-Lin1 becomes superior within the second forecast hour, while LK-Lin1 performs better in the first forecast hour. Only in the "twisted" category do LK-Lin1 and, even more, LK-Lin4 outperform DIS-Lin1 across all lead times. It should be noted, though, that the overall skill in the twisted category is very low for all competing models. In the "winding" category, LK-Lin1 slightly outperforms LK-Lin4 in the first 20 minutes. Finally, DIS-Rot1 performs worst at all lead times in all categories.

The change of model skill with lead time should be interpreted with care, as it depends on both the performance of the extrapolation model itself and the location error of the persistence model. For most models and SI categories, the skill appears to reach an optimum at some lead time, which implies that the superiority of the model over persistence reaches a maximum.

4. Conclusions

In this paper, we have introduced a framework to isolate and quantify the location error in precipitation nowcasts that are based on field-tracking techniques. While it is often assumed that errors in precipitation nowcasts are dominated by the temporal dynamics of precipitation intensity, the location error of predicted precipitation features has so far not been explicitly and formally quantified.

The main idea of our framework is to detect and track scale-invariant precipitation features (corners) in radar images. In our study, we detected features by using the approach of Shi and Tomasi (1994) and tracked these features following the approach of Lucas and Kanade [9], using both algorithms as implemented in the OpenCV library. We increased the robustness of extracted feature tracks by making sure that the features can be successfully tracked forwards and backwards. That approach, together with a rather strict definition of parameter values for feature detection and tracking, increases our confidence in the reliability of the detected tracks. Still, we have to assume that the feature locations themselves are, as any measurement, uncertain. We expect the main sources of uncertainty to be the grid resolution (which does not allow resolving errors below 1 km), and complex small-scale intensity dynamics that can interfere with motion patterns. For future studies, we suggest a comprehensive sensitivity analysis with regard to the parameters of the feature detection and tracking algorithms in order to better understand the effects on both the number and the robustness of detected tracks in the context of rainfall motion analysis. Still, we assume that the error of extrapolating feature motion is substantially larger than the error of feature tracking itself. In summary, we consider it warranted to use the observed tracks as a reference in order to evaluate the performance (or, inversely, the error) of any model that aims to predict the future locations of such precipitation features. For that purpose, we defined the location error of a forecast at any lead time Δt ahead of the forecast time t as the Euclidean distance between the observed and the predicted feature locations at $t + \Delta t$.

One might want to use this approach to comprehensively quantify the location error of any forecast model for the full spatial domain of a forecast grid, for example, a national radar composite. In such a case, we would need to assume that the average of forecast errors that we have quantified from observed feature locations in a forecast domain is representative for the average error of *all* location predictions in that domain. We have not yet investigated the validity of that assumption. One might argue that the

behavior of locations identified as "corners" or "good features to track" might not be representative for the motion behavior of the entire precipitation field; however, it will be difficult to find evidence to either verify or falsify such a hypothesis, as it would require another independent way to quantify the location error. Still, we are convinced that the proposed framework is useful: even without the need of strong assumptions on representativeness, the framework allows us to compare and benchmark the ability of different models to forecast future locations of precipitation features and thus to specifically focus on improving that ability by future model development.

The hypothesis that such further model developments are urgently required is supported by the results of our benchmarking study. It should be clarified again that this benchmark study does not intend to suggest better extrapolation models but to demonstrate the ability of our framework to unravel the location errors that are produced by state-of-the-art extrapolation methods. For that purpose, we compared four models: two models use the feature locations before and at forecast time t in order to derive displacement vectors which are then used to linearly extrapolate feature movement over the lead time. Model *LK-Lin*1 uses the feature locations at t and $t-1$, and *LK-Lin*4 uses the feature locations at t and $t-4$. The other two models are based on the dense optical flow algorithm DIS that generates a full motion vector field under various smoothness constraints. The model *DIS-Lin*1 obtains the displacement vector for a feature at t from the nearest motion vector in the field based on the radar images at times t and $t-1$ and uses that vector over the entire lead time. *DIS-Rot*1, in contrast, uses a Semi-Lagrangian scheme in which the displacement vector is updated as the feature moves through the motion field obtained from the DIS technique. The motivation behind the *DIS-Rot1* model is to better represent rotational or curved motion patterns. From these four competing models, LK-Lin4 appears to be the best model in the first forecast hour and DIS-Lin1 the best in the second. DIS-Rot1 performs consistently the worst. That is not quite in line with our naive expectation in which we would hope that a Semi-Lagrangian approach should be able to better capture at least curved motion patterns. But not even in the winding category does the complexity of the DIS-Rot1 approach pay off. Whether that is due to the implementation of the Semi-Lagrangian approach or due to the lack of validity of the approach should be the subject of future research. Comparing LK-Lin1 to LK-Lin4, we see a clear advantage in looking back in time more than one step. It appears that, this way, we can retrieve more reliable, more representative, and less noisy displacement vectors, which shows in the superiority of LK-Lin4 over LK-Lin1.

For all competing models, the mean location error exceeds a distance of 5 kilometers after 60 minutes and 10 kilometers after 110 minutes. At least 25% of all forecasts exceed an error of 5 kilometers after 50 minutes and an error of 10 kilometers after 90 minutes. Even for the best models in our experiment, at least 5 percent of the forecasts will have a location error of more than 10 kilometers after 45 minutes. When we relate such errors to application scenarios that are typically suggested for precipitation nowcasting, for example, in the context of early warning systems for pluvial floods in urban environments (see [19]), it becomes obvious that location errors matter: the order of magnitude of these errors is about the same as the typical extent of a convective cell or of a medium-sized city. Hence, the uncertainty of precipitation nowcasts at such length scales—just as a result of locational errors—can be substantial already at lead times of less than an hour.

While similar conclusions have already been drawn by using spatially sensitive verification measures such as the Fractions Skill Score (see, e.g., [6]), our framework allows us to isolate the location error for specific models and situations, to better understand the factors that govern these errors, and hence to use that knowledge in order to specifically improve the extrapolation of motion patterns in existing nowcasting models. As an example, we have demonstrated how the use of the sinuosity index can help us to better understand the predictive skill and hence the uncertainty of our models in specific situations. We hope that the large number of extracted tracks will help to foster the development of new techniques that use data-driven machine learning models for the extrapolation of feature location. For that purpose, we have made openly available the full set of extracted feature tracks for the year 2016 (https://doi.org/10.5281/zenodo.4024272 [20]) to serve as input to future studies. However, such future studies should also use radar data from a longer time period in order to learn more about the seasonal effects related to the properties of feature tracks.

Data Availability

The radar data are provided by DWD at https://opendata.dwd.de/weather/radar/radolan/ry (last access: Sept. 2020). The code of this analysis is available in the Github repository under https://github.com/arthurcts/loc_error (last access: Sept. 2020). The dataset of extracted feature tracks has been deposited in the Zenodo repository (https://doi.org/10.5281/zenodo.4024272).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors acknowledge the German Weather Service, namely, Dr. Tanja Winterrath, for making the RY data available from the latest RADKLIM reanalysis. Arthur Costa Tomaz de Souza has been funded by a Ph.D. scholarship of the German Academic Exchange Service (DAAD). Georgy Ayzel was partly funded by the ClimXtreme project (BMBF, FKZ 01LP1903B).

References

[1] M. Reyniers, *Quantitative Precipitation Forecasts Based on Radar Observations: Principles, Algorithms and Operational Systems*, Institut Royal Météorologique de Belgique, Brussel, Belgium, 2008.

[2] G. Ayzel, M. Heistermann, and T. Winterrath, "Optical flow models as an open benchmark for radar-based precipitation

nowcasting (rainymotion v0. 1)," *Geoscientific Model Development*, vol. 12, pp. 1387–1402, 2019.

[3] C. Pierce, A. Seed, S. Ballard, D. Simonin, and Z. Li, "Nowcasting," in *Doppler Radar Observations—Weather Radar, Wind Profiler, Ionospheric Radar, and Other Advanced Applications*, J. Bech, Ed., InTech, London, UK, 2012, http://www.intechopen.com/books/doppler-radar-observations-weather-radar-wind-profiler-ionospheric-radar-and-other-advanced-applications/nowcasting.

[4] M. E. Baldwin and J. S. Kain, "Sensitivity of several performance measures to displacement error, bias, and event frequency," *Weather and Forecasting*, vol. 21, no. 4, pp. 636–648, 2006.

[5] E. E. Ebert, "Fuzzy verification of high-resolution gridded forecasts: a review and proposed framework," *Meteorological Applications*, vol. 15, no. 1, pp. 51–64, 2008.

[6] G. Ayzel, T. Scheffer, and M. Heistermann, "RainNet v1.0: a convolutional neural network for radar-based precipitation nowcasting," *Geoscientific Model Development*, vol. 13, no. 6, pp. 2631–2644, 2020.

[7] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 151–172, 2000.

[8] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of the 9th IEEE Conference on Computer Vision and Pattern Recognition*, Springer, Seattle, WA, USA, 1994.

[9] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, p. 674, Vancouver, BC, Canada, August 1981.

[10] J. Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm," Technical report, Intel Corporation, Microprocessor Research Labs, Santa Clara, CA, USA, 2000.

[11] OpenCV library, "OpenCV: optical flow," 2020, https://docs.opencv.org/4.4.0/d4/dee/tutorial_optical_flow.html.

[12] J. E. Mueller, "An introduction to the hydraulic and topographic sinuosity Indexes1," *Annals of the Association of American Geographers*, vol. 58, no. 2, pp. 371–385, 1968.

[13] J. P. Terry and C.-C. Feng, "On quantifying the sinuosity of typhoon tracks in the western North Pacific basin," *Applied Geography*, vol. 30, no. 4, pp. 678–686, 2010.

[14] OpenCV library, "OpenCV: DISOpticalFlow class reference," 2020, https://docs.opencv.org/4.4.0/de/d4f/classcv_1_1DISOpticalFlow.html.

[15] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool, "Fast optical flow using dense inverse search," in *Proceedings of the European Conference on Computer Vision*, Amsterdam, The Netherlands, Springer, October 2016.

[16] U. Germann and I. Zawadzki, "Scale-dependence of the predictability of precipitation from continental radar images. Part I: description of the methodology," *Monthly Weather Review*, vol. 130, no. 12, pp. 2859–2873, 2002.

[17] T. Winterrath, "Erstellung einer radargestützten niederschlagsklimatologie (creation of a radar-based precipitation climatology)," *Berichte des Deutschen Wetterdienstes*, Deutscher Wetterdienst, Offenbach, Germany, 2017, https://www.dwd.de/DE/leistungen/pbfb_verlag_berichte/pdf_einzelbaende/251_pdf.pdf.

[18] DWD, "German climate Atlas," 2020, https://www.dwd.de/EN/ourservices/germanclimateatlas/germanclimateatlas.html.

[19] A. Zanchetta and P. Coulibaly, "Recent advances in real-time pluvial flash flood forecasting," *Water*, vol. 12, no. 2, p. 570, 2020.

[20] A. C. T. Souza, "Set of extracted feature tracks for the year 2016," Zenodo, 2020.