Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2018-166
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 10 September 2018

Geoscientific Model Development
Discussions
Open Access

Optical flow models as an open benchmark for radar-based precipitation nowcasting (rainymotion v0.1)

Georgy Ayzel[1], Maik Heistermann[1], and Tanja Winterrath[2]

[1]Institute for Earth and Environmental Sciences, University of Potsdam, Potsdam, Germany
[2]Deutscher Wetterdienst, Department of Hydrometeorology, Offenbach, Germany

Correspondence: Georgy Ayzel (ayzel@uni-potsdam.de)

Abstract. Quantitative precipitation nowcasting (QPN) has become an essential technique in various application contexts, such as early warning or urban sewage control. A common heuristic prediction approach is to track the motion of precipitation features from a sequence of weather radar images, and then to extrapolate that motion to the imminent future (minutes to hours), assuming that the intensity of the features remains constant ("Lagrangian persistence"). In that context, "optical flow" has become one of the most popular tracking techniques. Yet, the present landscape of computational QPN models still struggles with producing open software implementations. Focusing on this gap, we have developed and extensively benchmarked a stack of models based on different optical flow algorithms for the tracking step, and a set of parsimonious extrapolation procedures based on image warping and advection. We demonstrate that these models provide skillful predictions comparable with or even superior to state-of-the-art operational software. Our software library ("rainymotion") for precipitation nowcasting is written in Python programming language, and openly available at GitHub (github.com/hydrogo/rainymotion). That way, the library may serve as a tool for providing fast, free and transparent solutions that could serve as a benchmark for further model development and hypothesis testing – a benchmark that is far more advanced than the conventional benchmark of Eulerian persistence commonly used in QPN verification experiments.

1 Introduction

How much will it rain within the next hour? The term "quantitative precipitation nowcasting" refers to forecasts at high spatio-temporal resolution (60-600 seconds, 100-1000 meters) and short lead times of only a few hours. Nowcasts have become important for broad levels of the population for planning various kinds of activities. Yet, they are particularly relevant in the context of early warning of heavy convective rainfall events, and their corresponding impacts such as flash floods, landslides, or sewage overflow in urban areas.

While recent advances in numerical weather prediction (NWP) allow us to forecast atmospheric dynamics at very high resolution (Bauer et al., 2015), computational costs are typically prohibitive for the requirements of operational nowcasting applications with frequent update cycles. Furthermore, the heuristic extrapolation of rain field motion and development, as observed by weather radar, still appears to outperform NWP forecasts at very short lead times. Today, many precipitation nowcasting systems are operational at regional or national scales, utilizing various radar products, algorithms, and blending techniques in order to provide forecasts up to 1-3 hours: ANC (Mueller et al., 2003), MAPLE (Germann and Zawadzki, 2002), RADVOR (Winterrath et al., 2012), STEPS (Bowler et al., 2006), STEPS-BE (Foresti et al., 2016), SWIRLS (Cheung and Yeung, 2012; Woo and Wong, 2017). For an extensive review of existing operational systems, please refer to Reyniers (2008).

There are three main groups of techniques for radar-based precipitation nowcasting: analog, local Lagrangian, and stochastic (Foresti et al., 2016). Analog-based methods utilize a measure of similarity (e.g., correlation coefficient) to find the most similar sequences in archived radar observations in order to construct a precipitation forecast (Foresti et al., 2015). Stochastic nowcasts add random perturbations to deterministic ones to account for the uncertainty (Bowler et al., 2006; Foresti et al., 2016). In this study, we focus on the second group – local Lagrangian – which allows the extrapolation of the most recent radar images under the assumption that the velocity field is persistent (Germann and Zawadzki, 2002; Woo and Wong, 2017).

Local Lagrangian methods consist of two computational steps: tracking and forecasting (extrapolation) (Austin and Bellon, 1974). In the tracking step, we compute a velocity field from a series of consecutive radar images, either on a per pixel basis (Germann and Zawadzki, 2002; Grecu and Krajewski, 2000; Liu et al., 2015; Zahraei et al., 2012), or for contiguous objects (Zahraei et al., 2013). In the second step, we use that velocity field to advect the most recent rain field, i.e. to extrapolate its motion into the imminent future. That step has been implemented based on semi-Lagrangian schemes (Germann and Zawadzki, 2002), interpolation procedures (Liu et al., 2015), or mesh-based models (Bellerby, 2006; Zahraei et al., 2012). Different algorithms can be used for each step, tracking and forecasting, in order to compute an ensemble forecast (Grecu and Krajewski, 2000; Foresti et al., 2016).

One of the most prominent techniques for the tracking step is referred to as "optical flow". The original term was inspired by the idea of an *apparent* motion of brightness patterns observed when a camera or the eyeball is moving relative to the objects (Horn and Schunck, 1981). Today, optical flow is often understood as a group of techniques to infer motion patterns or velocity fields from consecutive image frames, e.g. in the field of precipitation nowcasting (Bowler et al., 2004; Liu et al., 2015; Woo and Wong, 2017). For the velocity field estimation, we need to accept both the brightness constancy assumption and one of a set of additional optical flow constraints (OFC). The spatial attribution of OFC marks the two main categories of optical flow models: local (differential) and global (variational) (Cheung and Yeung, 2012; Liu et al., 2015). Local models try to set an OFC only in some neighborhood, while global models apply an OFC for a whole image. Bowler et al. (2004) introduced the first local optical flow algorithm for precipitation nowcasting and gave rise to a new direction of models. Bowler's algorithm is the basis of the STEPS (Bowler et al., 2006) and STEPS-BE (Foresti et al., 2016) operational nowcasting systems. Liu et al. (2015) proposed using a local Lucas–Kanade optical flow method (Lucas and Kanade, 1981) independently for every pixel of satellite imagery and compared its performance with a global Horn–Schunck (Horn and Schunck, 1981) optical flow algorithm. Yeung et al. (2009), Cheung and Yeung (2012), and Woo and Wong (2017) used different global optical flow algorithms (Bruhn et al., 2005; Wong et al., 2009) for establishing the SWIRLS product for operational nowcasting in Hong-Kong.

Hence, for around two decades, optical flow algorithms have been doing their best for state-of-the-art operational nowcasting systems around the globe. Should research still care about them? It should... and the reason is that – despite the abundance of publications about different flavours of optical flow techniques for nowcasting applications – an open and transparent benchmark model is yet not available, except the most trivial one: Eulerian persistence. That is all the more surprising since open source libraries such as OpenCV (opencv.org) have been around for almost 20 years, providing efficient implementations of various optical flow algorithms for an endless number of research and application contexts. The aim of this paper is thus to establish a set of benchmark procedures for quantitative precipitation nowcasting as an alternative to Eulerian persistence. This study does not aim to improve the standard of precipitation nowcasting beyond the state-of-the-art, but to provide an open, transparent, reproducible and easy-to-use approach that can compete with the state-of-the-art, and against which future advances can be measured. To that end, we developed a group of models that are based on two optical flow formulations algorithms for the tracking step – local (Lucas and Kanade, 1981) and global (Farnebäck, 2003) – together with two parsimonious extrapolation techniques based on image warping and spatial interpolation. These models are verified against Eulerian persistence, as a trivial benchmark, and against the operational nowcasting system of the Deutscher Wetterdienst (the German Weather Service, DWD), as a representative of state-of-the-art models. The different optical flow implementations are published as an open source Python library (*rainymotion*, github.com/hydrogo/rainymotion) that entirely relies on free and open source dependencies, including detailed documentation and example workflows (rainymotion.readthedocs.io).

The paper is organized as follows. In Section 2, we describe the algorithmic and technical aspects of the suggested optical flow models. Section 3 describes the data we used, and provides a short synopsis of events we used for the benchmark experiment. We report the results in Section 4, and discuss them in various contexts in Section 5. Section 6 provides summary and conclusions.

2 Models

The benchmark models developed in this study consist of different combinations of algorithms for the two major steps of Lagrangian nowcasting frameworks, namely tracking and extrapolation (Austin and Bellon, 1974). Table 1 provides an overview of the models. The values of model parameters adopted in the benchmark experiment have been heuristically determined and not yet been subject to systematic optimization. For a description of parameters, please refer to Section S1 in the Supplementary or the *rainymotion* library documentation (rainymotion.readthedocs.io/).

2.1 Local optical flow models (the Sparse group)

The central idea around this group of methods is to identify distinct features in a radar image that are suitable for tracking. In this context, a "feature" is defined as a distinct point ("corner") with a sharp gradient of rainfall intensity. That approach is less arbitrary and scale dependent and thus more universal than classical approaches that track storm cells as contiguous objects (e.g., Wilson et al., 1998). Inside this group, we developed two models that slightly differ with regard to both tracking and extrapolation.

The first model (SparseSD, for Sparse Single Delta) uses only the two most recent radar images for identifying, tracking, and extrapolating features. Assuming that *t* denotes both the nowcast issue time and the time of the most recent radar image, the implementation can be summarized as follows:

1. Identify features in a radar image at time *t-1* using the Shi–Tomasi corner detector (Shi and Tomasi, 1994);

2. Track these features at time *t* using the local Lucas–Kanade optical flow algorithm (Lucas and Kanade, 1981);
3. Linearly extrapolate the features' motion in order to predict the features' locations at each lead time *n*;
4. Calculate the affine transformation matrix (Schneider and Eberly, 2003) for each lead time *n* based on the features' locations at time *t* and *t*+*n*;
5. Warp the radar image at time *t* for each lead time *n* using the corresponding affine matrix, and linearly interpolate remaining discontinuities (Wolberg, 1990).

For a visual representation of the SparseSD model routine, please refer to Fig. 1.

The second model (Sparse) uses the 24 recent radar images, and we consider here only features that are persistent over the whole period (of 24 timesteps) for capturing the most steady movement. Its implementation can be summarized as follows:

1. Identify features on a radar image at time *t-23* using the Shi–Tomasi corner detector (Shi and Tomasi, 1994);
2. Track these features on radar images at the time from *t-22* to *t* using the local Lucas–Kanade optical flow algorithm (Lucas and Kanade, 1981);
3. Build linear regression models which independently parametrize changes in coordinates through time (from *t-23* to *t*) for every successfully tracked feature;
4. Continue with steps 3-5 of SparseSD.

For a visual representation of the Sparse model routine, please refer to Fig. 2.

To our knowledge, this study is the first to apply image warping directly as a simple and fast algorithm to represent advective motion of a precipitation field. In Section S2 of the Supplementary, you can find a simple synthetic example which shows the potential of the warping technique to replace an explicit advection formulation for temporal extrapolation.

2.2 Global optical flow models (the Dense group)

The Dense group of models uses the global optical flow algorithm proposed by Farnebäck (2003) which allows us to explicitly estimate the velocity of each image pixel based on an analysis of two consecutive radar images. The two models in this group differ only with regard to the extrapolation (or advection) step. The first model (Dense) uses a constant-vector advection scheme (Bowler et al., 2004), while the second model (DenseRotation) uses a forward semi-Lagrangian advection scheme (Germann and Zawadzki, 2002). The main difference between the proposed approaches is that a constant-vector scheme does not allow for the representation of rotational motion (Bowler et al., 2004); a semi-Lagrangian scheme allows for large-scale rotational movement while assuming the motion field itself to be persistent (Fig. 3). Both models utilize a linear interpolation procedure in order to interpolate advected rainfall intensities at their predicted locations to the original radar grid. This interpolation procedure follows the same idea of distributing the value of a rain pixel to its neighborhood, as proposed in different modifications by Bowler et al. (2004), Liu et al. (2015), and Zahraei et al. (2012). The Dense group models' implementation can be summarized as follows:

- Calculate a continuous displacement field using a global Farnebäck optical flow algorithm (Farnebäck, 2003) based on the radar images at time *t-1* and *t*;

- Use a constant-vector (Bowler et al., 2004) or a semi-Lagrangian scheme (Germann and Zawadzki, 2002) to extrapolate (advect) each pixel according to the obtained displacement (velocity) field, in one single step for each lead time *t+n*. For the semi-Lagrangian scheme, we update the velocity of each displaced pixel at each prediction time step by retrieving the velocity closest to the predicted pixel location;

- As a result of the advection step, we basically obtain an irregular point cloud that consists of the original radar pixels displaced from their original location. We use the intensity of each displaced pixel at its predicted location at time *t+n* in order to interpolate the intensity at each grid point of the original (native) radar grid (Liu et al., 2015; Zahraei et al., 2012). It is important to note that we minimize numerical diffusion by first advecting each pixel over the target lead time before applying the interpolation procedure. That way, we avoid rainfall features to be smoothed in space by the effects of interpolation.

2.3 Persistence

The (trivial) benchmark model of Eulerian persistence assumes that for any lead time *n*, the precipitation field is the same as for time *t*. Considering its simplicity, it is quite a powerful predictor for very short lead times, and, at the same time, its verification performance is a good measure of temporal decorrelation for different events.

2.4 *rainymotion* Python library

We have developed the *rainymotion* Python library that implements the above models. The source code is available in a Github repository (github.com/hydrogo/rainymotion), and has a documentation page (rainymotion.readthedocs.io) which includes installation instructions, models' description, and usage examples. The library code and accompanying documentation are freely distributed under the MIT software license which allows unrestricted use. The library is written in the Python 3 programming language (python.org) and its core is entirely based on open source software libraries (Fig. 4): *OpenCV* (Bradski and Kaehler, 2008), *SciPy* (Jones et al., 2018), *NumPy* (Oliphant, 2006), *Scikit-learn* (Pedregosa et al., 2011), and *Scikit-image* (Van der Walt et al., 2014). For manipulation of the data stored in HDF databases we also use the *h5py* library (h5py.org), to generate figures we use the *Matplotlib* library (Hunter, 2007), and we use the *Jupyter notebook* (jupyter.org) interactive development environment for code and documentation development and distribution. For managing the dependencies without any conflicts, we recommend to use the *Anaconda* Python distribution (anaconda.com) and follow *rainymotion* installation instructions (rainymotion.readthedocs.io/en/latest/gettingstarted.html#installation).

2.5 Operational baseline (RADVOR)

The DWD operationally runs a stack of models for radar-based nowcasting providing precipitation forecasts for a lead time up to 2 hours. The operational quantitative precipitation nowcasts are based on the RADVOR module (Bartels et al., 2005; Rudolf et al., 2012). The tracking algorithm estimates the motion field from the latest sequential clutter-filtered radar images using a pattern recognition technique on different spatial resolutions (Winterrath and Rosenow, 2007; Winterrath et al., 2012). The focus of the tracking algorithm is on the meso-β scale (spatial extent: 25–250 km) to cover mainly large-scale precipitation patterns, but the meso-γ scale (spatial extension: 2.5–25 km) is also incorporated to allow the detection of smaller-scale convective structures. The resulting displacement field is interpolated to a regular grid and a weighted averaging with previously derived displacement fields is implemented to guarantee a smooth displacement over time. The extrapolation of the most recent radar image according to the obtained velocity field is performed using a semi-Lagrangian approach. The described operational model is updated every 5 minutes and produces precipitation nowcasts at a temporal resolution of 5 minutes and a lead time of 2 hours (RV product). In the presented study we used the RV product data as an operational baseline and did not implement a programmatic realization of the DWD algorithm itself.

2.6 Verification

For the verification we use two general categories of scores: continuous (based on the differences between nowcast and observed rainfall intensities) and categorical (based on standard contingency tables for calculating matches between boolean values which reflect the exceedance of specific rainfall intensity thresholds). We use the mean absolute error (MAE) as a continuous score:

$$MAE = \frac{\sum_{i=1}^{n} |now_i - obs_i|}{n} \tag{1}$$

where now_i and obs_i are nowcast and observed rainfall rate in the *i*-th pixel of the corresponding radar image, and *n* the number of pixels.

And we use the critical success index (CSI) as a categorical score:

$$CSI = \frac{hits}{hits + false\ alarms + misses} \tag{2}$$

where *hits*, *false alarms*, and *misses* are defined by the contingency table and the corresponding threshold value (for details see Section S4 of the Supplementary).

Following the study of Bowler et al. (2006) we have applied threshold rain rates of 0.125, 0.25, 0.5 and 1 mm h^{-1} for calculating the CSI.

These two metrics inform us about the models' performance from the two perspectives: MAE captures errors in rainfall rates prediction (the less the better), and CSI captures model accuracy (the fraction of the forecast event that was correctly predicted; does not distinguish the source of errors; the higher the better). You can find results represented in terms of additional categorical scores (false alarm rate, probability of detection, equitable threat score) in Section S4 of the Supplementary.

3 Radar data and verification events

We use the so-called RY product of the DWD as input to our nowcasting models. The RY product represents a quality-controlled rainfall depth product that is a composite of the 17 operational Doppler radars maintained by the DWD. It has a spatial extent of 900×900 km and covers the whole area of Germany. Spatial and temporal resolution of the RY product is 1×1 km and 5 minutes, respectively. This composite product includes various procedures for correction and quality control (e.g. clutter removal). We used the $\omega radlib$ (Heistermann et al., 2013) software library for reading the DWD radar data.

For the analysis, we have selected 11 events during the summer periods of 2016 and 2017. These events are selected for covering a range of event characteristics with different rainfall intensity, spatial coverage, and duration. Table 2 shows the studied events. You can also find links to animations of event intensity dynamics in Section S3 of the Supplementary.

4 Results

For each event, all models (Sparse, SparseSD, Dense, DenseRotation, Persistence) were used to compute nowcasts with lead times from 5 to 60 minutes (in 5 minute steps). Operational nowcasts generated by the RADVOR system were provided by the DWD with the same temporal settings.

Figure 5 shows the model performance (in terms of MAE) as a function of lead time. For each event, the Dense group of models is superior to the other ones. The RV product achieves an average rank between models of the Sparse and Dense groups. The SparseSD model outperforms the Sparse model for short lead times (up to 10-15 minutes), and vice versa for longer lead times. For some events (1-4, 6, 10, 11), the performance of the RV product appears to be particularly low in the first 10 minutes, compared to the other models. These events are characterized by particularly fast rainfall field movement.

Figure 6 has the same structure as Fig. 5, but shows the CSI with a threshold value of 0.125 mm h^{-1}. For two events (7 and 10) the RV product performs better than optical flow based models for lead times beyond 30 minutes, and the Sparse group outperforms the Dense group for lead times beyond 45 minutes. For the remaining events, the Dense group outperforms all other methods. For the Dense group of models, it is clear that accounting for the rotation in the field only improves the forecast when strong rotation exists (e.g., event #7, which is consistent with the performance of the RV product that also follows a semi-Lagrangian approach). For the majority of events, however, the constant-vector advection scheme (the Dense model) appears to perform slightly better then the semi-Lagrangian scheme (the DenseRotation model). The behavior of the Sparse group models is mostly consistent with the MAE.

Figure 7 shows the model performance using the CSI with a threshold value of 1 mm h^{-1}. For the majority of events, the resulting ranking of models is the same as for the CSI with a threshold of 0.125 mm h^{-1}. For three events (2, 7, 10) the RV product performs better for lead times beyond 20-30 minutes.

Table 3 summarizes the results of the Dense group models in comparison to the RV model for different verification metrics averaged over all the selected events and two lead time periods: 5–30, and 35–60 minutes. Results show that the Dense group always slightly outperforms the RV model for lead times up to 30 minutes. For lead times from 30 minutes to 1 hour, differences between model performances are less pronounced. For the CSI metric, the differences between all models tend to decrease with increasing rainfall thresholds.

You can find more figures illustrating the models' efficiency for different thresholds and lead times in Section S4 of the Supplementary.

5 Discussion

5.1 Model comparison

For the analyzed events, there is a clear pattern that the Dense group of optical flow models outperforms the operational RV model for shorter lead times (up to 30 minutes) and sometimes (events 2, 7, 10) underperforms for longer lead times (from 30 minutes to 1 hour). That behavior of the RV product could be a result of accounting for meso-β scale features in the velocity field computation (Winterrath et al., 2012) which is designed to capture movement patterns at a larger scale, concerted with a weighted averaging of the derived displacement vectors over the three recent time steps (in order to guarantee a steady displacement over time). A gain in performance for longer lead times by taking into account more time steps from the past can also be observed when comparing the SparseSD model (looks back five minutes in time) against the Sparse model (looks back two hours in time).

The natural properties of the precipitation formation process limit predictability. Many studies specify a lead time of 30 minutes as a predictability limit for convective structures with fast dynamics of rainfall evolution (Foresti et al., 2016; Grecu and Krajewski, 2000; Thorndahl et al., 2017; Wilson et al., 1998; Zahraei et al., 2012). Our study confirms these findings.

There are a couple of possible directions for enhancing the performance for longer lead times using the Dense group of models. The first way is to adopt the RV scheme for a weighted averaging of calculated velocity fields separately derived for the last three (or more) time steps. Another option is to calculate separate velocity fields for low and high intensity subregions of the rain field, and advect these subregions separately (like proposed in Golding (1998)), or find an optimal weighting procedure. The third possible way is to implement different smoothness constraints when implementing optical flow algorithms like proposed in Germann and Zawadzki (2002), Bowler et al. (2004), or Mecklenburg et al. (2000). The flexibility of the *rainymotion* software library allows users to incorporate such algorithms for benchmarking any hypothesis, and e.g. implement different models or parameterisations for different lead times. Bowler et al. (2004) also showed a significant performance increase for longer lead times by using NWP model winds for the advection step. However, Winterrath and Rosenow (2007) did not obtain any improvement compared to RADVOR for longer lead times by incorporating NWP model winds in the nowcasting procedure.

5.2 Advection schemes properties and effectiveness

In our study, we have shown the advantages of the constant-vector advection scheme (implemented in the Dense model, provided in Bowler et al. (2004)) over the semi-Lagrangian scheme (implemented in the DenseRotation model). For the majority of events (except the event #7) and lead times it appears that a linear extrapolation together with a constant-vector advection are preferable, particularly in a context where we have a complex motion pattern with an absence of distinct large-scale rotation. However, when a precipitation field has a clear rotational component (e.g. counter-clock wise cyclonic rotation during the event #7), accounting for rotation increases the forecast efficiency (DenseRotation outperforms Dense). It is also possible that the positive effect of using a semi-Lagrangian advection scheme may be more evident for lead times longer than one hour, and for smoothed velocity fields (as shown in Germann and Zawadzki (2002)).

5.3 Computational performance

Computational performance might be an important criterion for end users aiming at update cycles with high frequency. We ran our nowcasting models on a standard office PC with an Intel® Core^{TM} i7-2600 CPU (8 cores, 3.4 GHz), and on a standard laptop with an Intel® Core^{TM} i5-7300HQ CPU (4 cores, 2.5 GHz). The average time for generating one nowcast for one hour lead time (at 5 minute resolution) for the Sparse group is 2-3 s, and for the Dense group is 150-180 s. The Dense group consumes more computational resources mostly because of expensive interpolation operation implemented for images of high resolution (900×900 pixels). There is also ample potential for increasing the computational performance of the interpolation. It might also be considered to combine the warping procedure for the extrapolation step with the Dense optical flow procedure for the tracking step in order to dramatically enhance computational performance. For that purpose, however, the errors introduced by the warping procedure need to be understood better.

6 Summary and conclusions

Optical flow is a technique for deriving a velocity field from consecutive series of images which is widely used in image analysis, and became increasingly popular in meteorological applications over the past 20 years. In our study, we examined the performance of optical flow based models for radar-based precipitation nowcasting, as implemented in the open-source *rainymotion* library, for a wide range of rainfall events using radar data provided by the DWD.

The comparison of the models' verification performance with the performance of the operational baseline (the RV product provided by the DWD) shows a firm basis for using optical flow in radar-based precipitation nowcasting studies. For the majority of the analyzed events, models which use the global optical flow algorithm for deriving a displacement vector field outperform the operational model. The Sparse group of models showed significant skill, yet they performed generally poorer than both the Dense group and the RV product. We should, however, not prematurely discard the group of Sparse models before we have not gained a better understanding of error sources with regard to the tracking, extrapolation and warping steps.

There is a clear and rapid model performance loss over lead time for events with high rainfall intensities. This issue continues to be unresolved by standard nowcasting approaches, but some improvement in this field may be achieved with using strategies such as nowcasts merging with NWP results and stochastic accounting of rainfall field evolution. We suppose that using new data-driven models based on machine and deep learning approaches may provide additional gain in performance by utilizing and structuring common patterns in the massive archives of radar data.

We do not claim that the developed models will compete with well-established and excessively tuned operational services for radar-based precipitation nowcasting, but we hope our models may serve as an essential tool for providing a fast, free and open source solution that can serve as a benchmark for further model development and hypothesis testing – a benchmark that is far more advanced than the conventional benchmark of Eulerian persistence.

Code and data availability. The *rainymotion* library is free and open source. It is distributed under the MIT software license which allows unrestricted use. The source code is provided through a GitHub repository github.com/hydrogo/rainymotion, and the documentation is available on a website http://rainymotion.readthedocs.io. The DWD provided the sample data of the RY product, and it is distributed with the *rainymotion* repository to provide a real case and reproducible example of precipitation nowcasting. We encourage any collaboration to make free and open source models for radar-based precipitation nowcasting better and accessible for a broader public.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. This publication was financially supported by Geo.X, the Research Network for Geosciences in Berlin and Potsdam.

References

Austin, G. L. and Bellon, A.: The use of digital weather radar records for short-term precipitation forecasting, Quarterly Journal of the Royal Meteorological Society, 100, 658–664, https://doi.org/10.1002/qj.49710042612, http://doi.wiley.com/10.1002/qj.49710042612, 1974.

Bauer, P., Thorpe, A., and Brunet, G.: The quiet revolution of numerical weather prediction, https://doi.org/10.1038/nature14956, http://www.nature.com/doifinder/10.1038/nature14956, 2015.

Bellerby, T. J.: High-resolution 2-D cloud-top advection from geostationary satellite imagery, IEEE Transactions on Geoscience and Remote Sensing, 44, 3639–3648, https://doi.org/10.1109/TGRS.2006.881117, http://ieeexplore.ieee.org/document/4014303/, 2006.

Bowler, N. E., Pierce, C. E., and Seed, A.: Development of a precipitation nowcasting algorithm based upon optical flow techniques, Journal of Hydrology, 288, 74–91, https://doi.org/10.1016/j.jhydrol.2003.11.011, https://www.sciencedirect.com/science/article/pii/S0022169403004591, 2004.

Bowler, N. E., Pierce, C. E., and Seed, A. W.: STEPS: A probabilistic precipitation forecasting scheme which merges an extrapolation nowcast with downscaled NWP, Quarterly Journal of the Royal Meteorological Society, 132, 2127–2155, https://doi.org/10.1256/qj.04.100, http://dx.doi.org/10.1256/qj.04.100, 2006.

Bradski, G. and Kaehler, A.: Learning OpenCV: Computer vision with the OpenCV library, " O'Reilly Media, Inc.", 2008.

Bruhn, A., Weickert, J., Feddern, C., Kohlberger, T., and Schnörr, C.: Variational optical flow computation in real time, IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 14, 608–15, https://doi.org/10.1109/TIP.2005.846018, https://ieeexplore.ieee.org/document/1420392/, 2005.

Cheung, P. and Yeung, H. Y.: Application of optical-flow technique to significant convection nowcast for terminal areas in Hong Kong, in: The 3rd WMO International Symposium on Nowcasting and Very Short-Range Forecasting (WSN12), pp. 1–10, http://www.hko.gov.hk/publica/reprint/r1025.pdf, 2012.

Farnebäck, G.: Two-frame motion estimation based on polynomial expansion, Image Analysis, 2003, 363–370, https://doi.org/10.1007/3-540-45103-x_50, http://link.springer.com/10.1007/3-540-45103-X, 2003.

Foresti, L., Panziera, L., Mandapaka, P. V., Germann, U., and Seed, A.: Retrieval of analogue radar images for ensemble nowcasting of orographic rainfall, Meteorological Applications, 22, 141–155, https://doi.org/10.1002/met.1416, http://doi.wiley.com/10.1002/met.1416, 2015.

Foresti, L., Reyniers, M., Seed, A., and Delobbe, L.: Development and verification of a real-time stochastic precipitation nowcasting system for urban hydrology in Belgium, Hydrology and Earth System Sciences, 20, 505–527, https://doi.org/10.5194/hess-20-505-2016, https://www.hydrol-earth-syst-sci.net/20/505/2016/, 2016.

Germann, U. and Zawadzki, I.: Scale-Dependence of the Predictability of Precipitation from Continental Radar Images. Part I: Description of the Methodology, Monthly Weather Review, 130, 2859–2873, https://doi.org/10.1175/1520-0493(2002)130<2859:SDOTPO>2.0.CO;2, http://journals.ametsoc.org/doi/abs/10.1175/1520-0493{%}282002{%}29130{%}3C2859{%}3ASDOTPO{%}3E2.0.CO{%}3B2, 2002.

Golding, B. W.: Nimrod: A system for generating automated very short range forecasts, Meteorological Applications, 5, 1–16, https://doi.org/10.1017/S1350482798000577, http://doi.wiley.com/10.1017/S1350482798000577, 1998.

Grecu, M. and Krajewski, W. F.: A large-sample investigation of statistical procedures for radar-based short-term quantitative precipitation forecasting, Journal of Hydrology, 239, 69–84, https://doi.org/10.1016/S0022-1694(00)00360-7, https://www.sciencedirect.com/science/article/pii/S0022169400003607, 2000.

Heistermann, M., Jacobi, S., and Pfaff, T.: Technical Note: An open source library for processing weather radar data (wradlib), Hydrology and Earth System Sciences, 17, 863–871, https://doi.org/10.5194/hess-17-863-2013, https://www.hydrol-earth-syst-sci.net/17/863/2013/, 2013.

Horn, B. K. and Schunck, B. G.: Determining optical flow, Artificial intelligence, 17, 185–203, 1981.

Hunter, J. D.: Matplotlib: A 2D graphics environment, Computing in science & engineering, 9, 90–95, 2007.

Jones, E., Oliphant, T., and Peterson, P.: SciPy: open source scientific tools for Python, https://scipy.org/, last access: 29 June 2018, 2018.

Liu, Y., Xi, D. G., Li, Z. L., and Hong, Y.: A new methodology for pixel-quantitative precipitation nowcasting using a pyramid Lucas Kanade optical flow approach, Journal of Hydrology, 529, 354–364, https://doi.org/10.1016/j.jhydrol.2015.07.042, https://www.sciencedirect.com/science/article/pii/S002216941500548X, 2015.

Lucas, B. D. and Kanade, T.: An iterative image Registration Technique with an Application to Stereo Vision, in: Proceedings DARPA Image Understanding Workrhop, pp. 674–679, Morgan Kaufmann Publishers Inc., https://doi.org/10.1145/358669.358692, https://ri.cmu.edu/pub_files/pub3/lucas_bruce_d_1981_2/lucas_bruce_d_1981_2.pdf, 1981.

Mecklenburg, S., Joss, J., and Schmid, W.: Improving the nowcasting of precipitation in an Alpine region with an enhanced radar echo tracking algorithm, Journal of Hydrology, 239, 46–68, https://doi.org/10.1016/S0022-1694(00)00352-8, https://www.sciencedirect.com/science/article/pii/S0022169400003528, 2000.

Mueller, C. M., Axen, T. S., Oberts, R. R., Ilson, J. W., Etancourt, T. B., Ettling, S. D., and Ien, N. O.: NCAR Auto-Nowcast System, Weather and Forecasting, 18, 545–561, https://doi.org/10.1175/1520-0434(2003)018<0545:NAS>2.0.CO;2, http://journals.ametsoc.org/doi/abs/10.1175/1520-0434{%}282003{%}29018{%}3C0545{%}3ANAS{%}3E2.0.CO{%}3B2, 2003.

Oliphant, T. E.: A guide to NumPy, vol. 1, Trelgol Publishing USA, 2006.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in Python, Journal of machine learning research, 12, 2825–2830, 2011.

Reyniers, M.: Quantitative precipitation forecasts based on radar observations: Principles, algorithms and operational systems, Institut Royal Météorologique de Belgique, https://www.meteo.be/meteo/download/fr/3040165/pdf/rmi_scpub-1261.pdf, 2008.

Schneider, P. J. and Eberly, D. H.: Geometric tools for computer graphics, Boston, https://www.sciencedirect.com/science/book/9781558605947, 2003.

Shi, J. and Tomasi, C.: Good features to track, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94, pp. 593–600, IEEE Comput. Soc. Press, https://doi.org/10.1109/CVPR.1994.323794, http://ieeexplore.ieee.org/document/323794/, 1994.

Thorndahl, S., Einfalt, T., Willems, P., Nielsen, J. E., ten Veldhuis, M.-C., Arnbjerg-Nielsen, K., Rasmussen, M. R., and Molnar, P.: Weather radar rainfall data in urban hydrology, Hydrology and Earth System Sciences, 21, 1359–1380, https://doi.org/10.5194/hess-21-1359-2017, http://www.hydrol-earth-syst-sci.net/21/1359/2017/, 2017.

Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., and Yu, T.: scikit-image: image processing in Python, PeerJ, 2, e453, https://doi.org/10.7717/peerj.453, https://peerj.com/articles/453/, 2014.

Wilson, J. W., Crook, N. A., Mueller, C. K., Sun, J., and Dixon, M.: Nowcasting Thunderstorms: A Status Report, Bulletin of the American Meteorological Society, 79, 2079–2099, https://doi.org/10.1175/1520-0477(1998)079<2079:NTASR>2.0.CO;2, https://journals.ametsoc.org/doi/abs/10.1175/1520-0477%281998%29079%3C2079%3ANTASR%3E2.0.CO%3B2, 1998.

Winterrath, T. and Rosenow, W.: A new module for the tracking of radar-derived precipitation with model-derived winds, Advances in Geosciences, 10, 77–83, https://doi.org/10.5194/adgeo-10-77-2007, http://www.adv-geosci.net/10/77/2007/, 2007.

Winterrath, T., Rosenow, W., and Weigl, E.: On the DWD quantitative precipitation analysis and nowcasting system for real-time application in German flood risk management, in: Weather Radar and Hydrology (Proceedings of a symposium held in Exeter, UK, April 2011) IAHS Publ., vol. 351, pp. 323–329, https://www.dwd.de/DE/leistungen/radolan/radolan_info/Winterrath_German_flood_risk_management_pdf.pdf?__blob=publicationFile&v=4, 2012.

Wolberg, G.: Digital Image Warping, IEEE Computer Society Press, 1990.

Wong, W. K., Yeung, L. H. Y., Wang, Y. C., and Chen, M.: Towards the blending of NWP with nowcast - Operation experience in B08FDP, in: WMO symposium on nowcasting, http://my.hko.gov.hk/publica/reprint/r844.pdf, 2009.

Woo, W.-C. and Wong, W.-K.: Operational Application of Optical Flow Techniques to Radar-Based Rainfall Nowcasting, Atmosphere, 8, 48, https://doi.org/10.3390/atmos8030048, http://www.mdpi.com/2073-4433/8/3/48, 2017.

Yeung, L. H. Y., Wong, W. K., Chan, P. K. Y., Lai, E. S. T., Yeung, L. H. Y., Wong, W. K., Chan, P. K. Y., and Lai, E. S. T.: Applications of the Hong Kong Observatory nowcasting system SWIRLS-2 in support of the 2008 Beijing Olympic Games, in: WMO Symposium on Nowcasting, vol. 30, http://my.hko.gov.hk/publica/reprint/r843.pdf, 2009.

Zahraei, A., lin Hsu, K., Sorooshian, S., Gourley, J. J., Lakshmanan, V., Hong, Y., and Bellerby, T.: Quantitative Precipitation Nowcasting: A Lagrangian Pixel-Based Approach, Atmospheric Research, 118, 418–434, https://doi.org/10.1016/j.atmosres.2012.07.001, https://www.sciencedirect.com/science/article/pii/S0169809512002219, 2012.

Zahraei, A., lin Hsu, K., Sorooshian, S., Gourley, J. J., Hong, Y., and Behrangi, A.: Short-term quantitative precipitation forecasting using an object-based approach, Journal of Hydrology, 483, 1–15, https://doi.org/10.1016/j.jhydrol.2012.09.052, https://www.sciencedirect.com/science/article/pii/S0022169412008694, 2013.

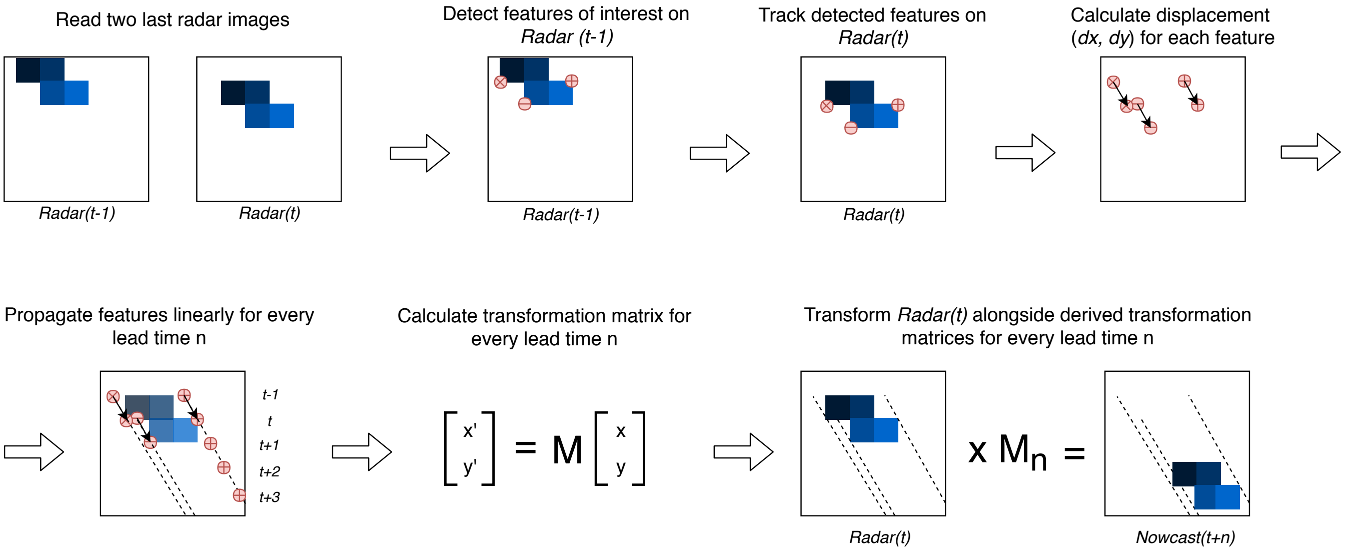
Read two last radar images
Radar(t-1)
Radar(t)
Detect features of interest on Radar (t-1)
Radar(t-1)
Track detected features on Radar(t)
Radar(t)
Calculate displacement (dx, dy) for each feature
Propagate features linearly for every lead time n
t-1
t
t+1
t+2
t+3
Calculate transformation matrix for every lead time n
[x' y'] = M [x y]
Transform Radar(t) alongside derived transformation matrices for every lead time n
Radar(t)
x M_n =
Nowcast(t+n)

Figure 1. Scheme of the SparseSD model

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2018-166
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 10 September 2018

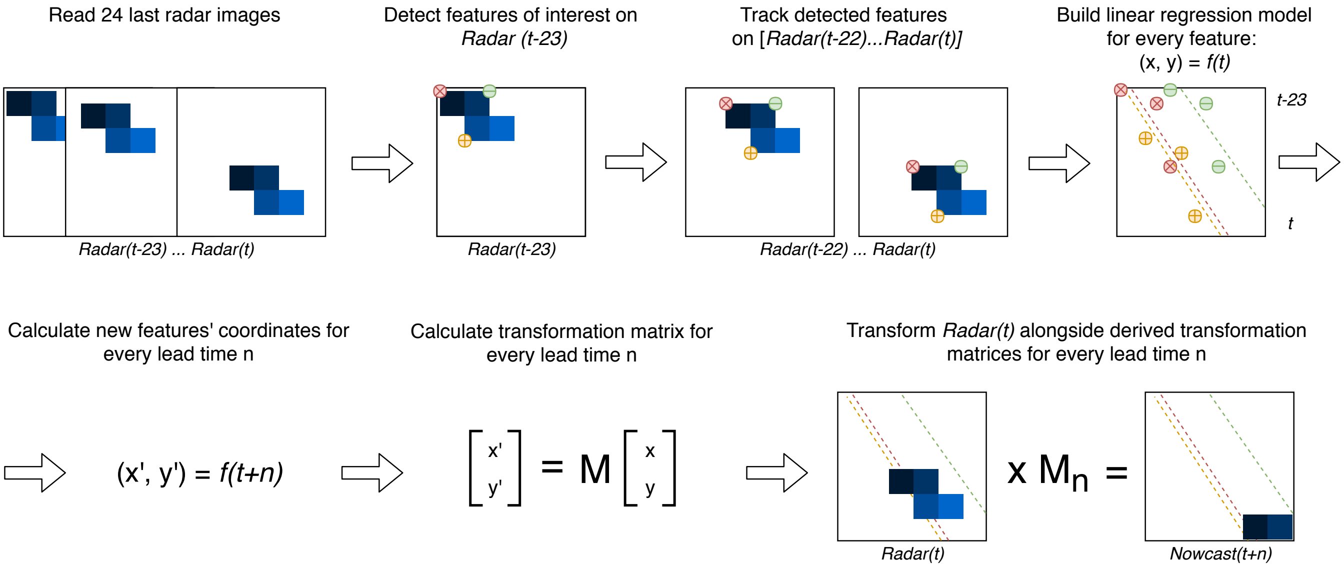
Read 24 last radar images
Radar(t-23) ... Radar(t)
Detect features of interest on Radar (t-23)
Radar(t-23)
Track detected features on [Radar(t-22)...Radar(t)]
Radar(t-22) ... Radar(t)
Build linear regression model for every feature: (x, y) = f(t)
t-23
t
Calculate new features' coordinates for every lead time n
(x', y') = f(t+n)
Calculate transformation matrix for every lead time n
[x' y'] = M [x y]
Transform Radar(t) alongside derived transformation matrices for every lead time n
x Mn =
Radar(t)
Nowcast(t+n)

Figure 2. Scheme of the Sparse model

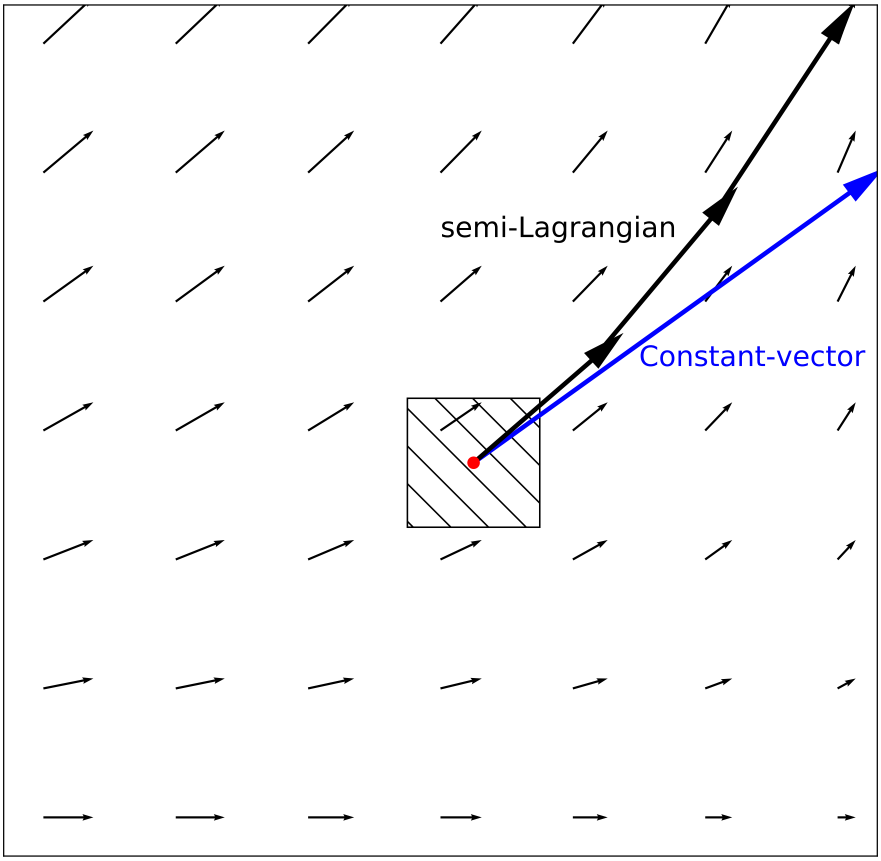

Figure 3. Displacement vectors of two proposed advection schemes: constant vector and semi-Lagrangian

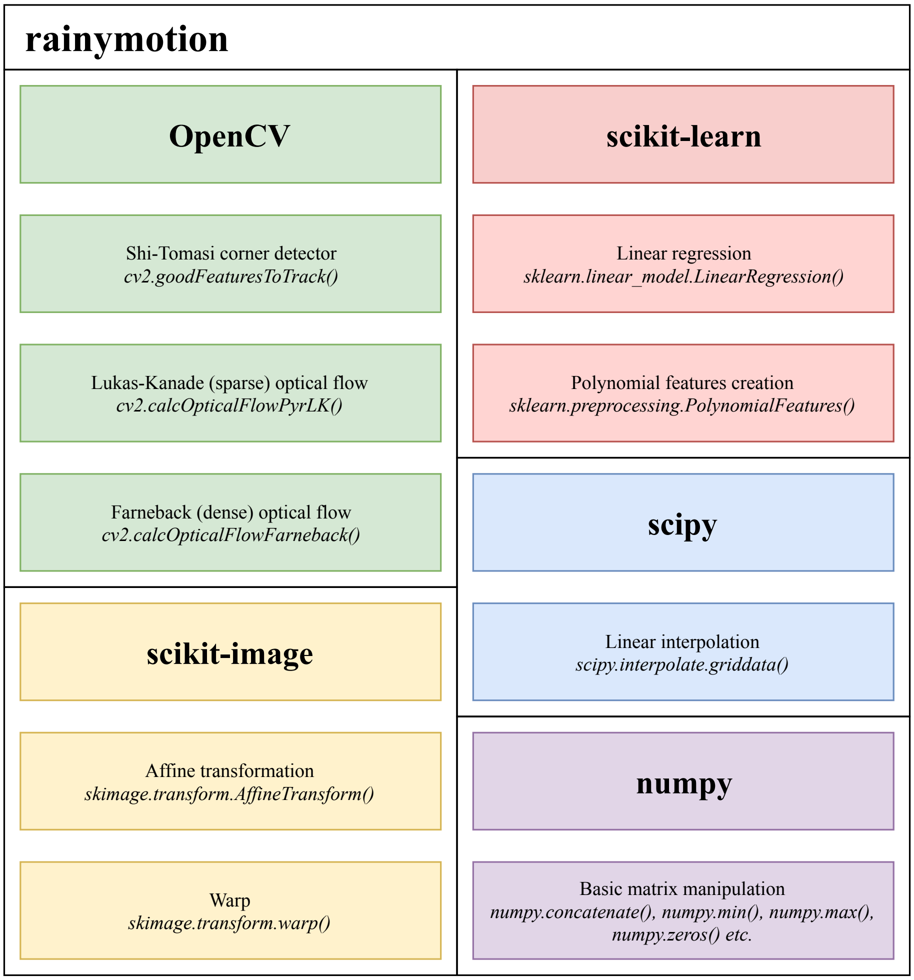
rainymotion
OpenCV
Shi-Tomasi corner detector
cv2.goodFeaturesToTrack()
Lukas-Kanade (sparse) optical flow
cv2.calcOpticalFlowPyrLK()
Farneback (dense) optical flow
cv2.calcOpticalFlowFarneback()
scikit-image
Affine transformation
skimage.transform.AffineTransform()
Warp
skimage.transform.warp()
scikit-learn
Linear regression
sklearn.linear_model.LinearRegression()
Polynomial features creation
sklearn.preprocessing.PolynomialFeatures()
scipy
Linear interpolation
scipy.interpolate.griddata()
numpy
Basic matrix manipulation
numpy.concatenate(), numpy.min(), numpy.max(), numpy.zeros() etc.

Figure 4. Key Python libraries for *rainymotion* library development

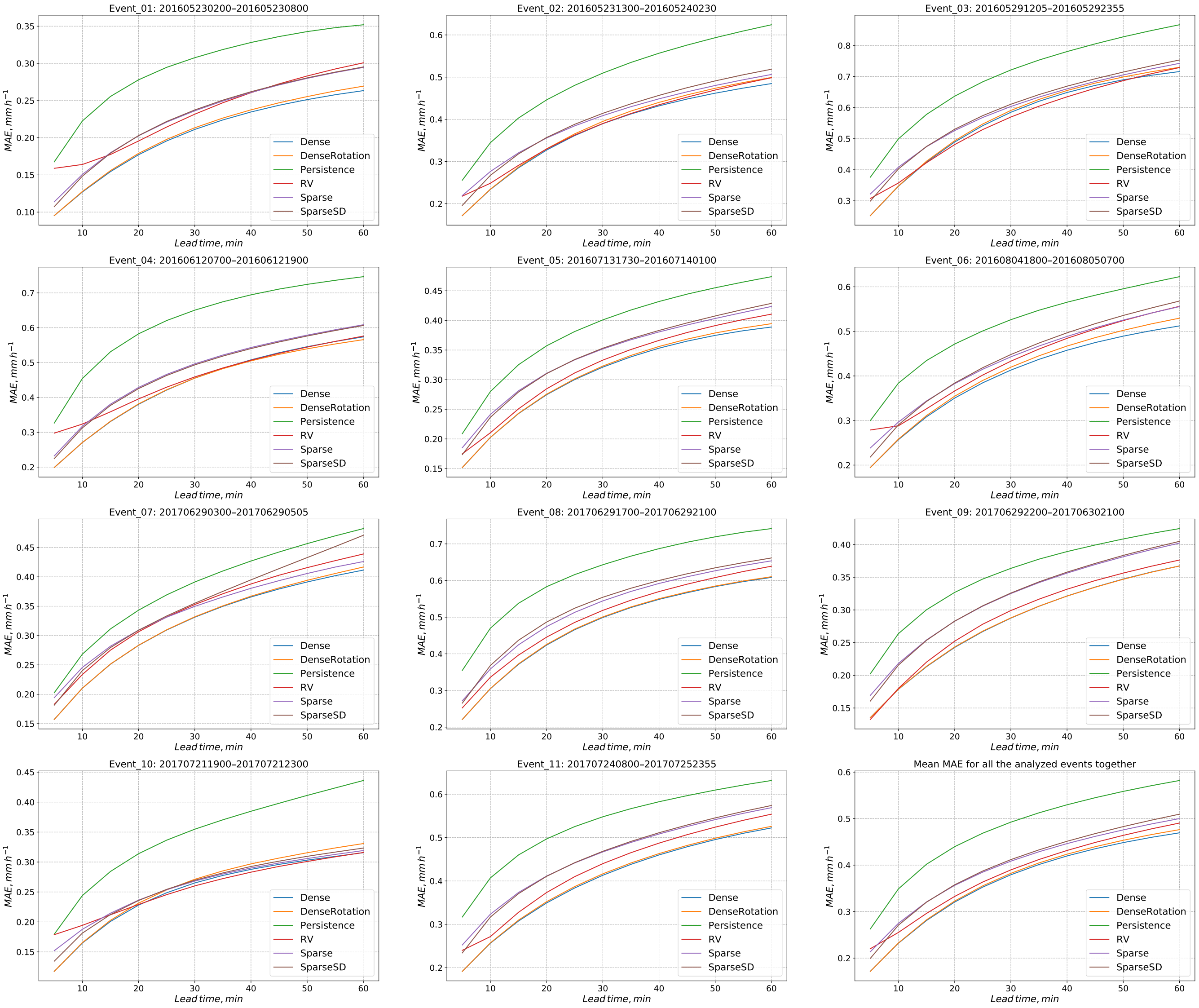

Figure 5. Lead time wise MAE

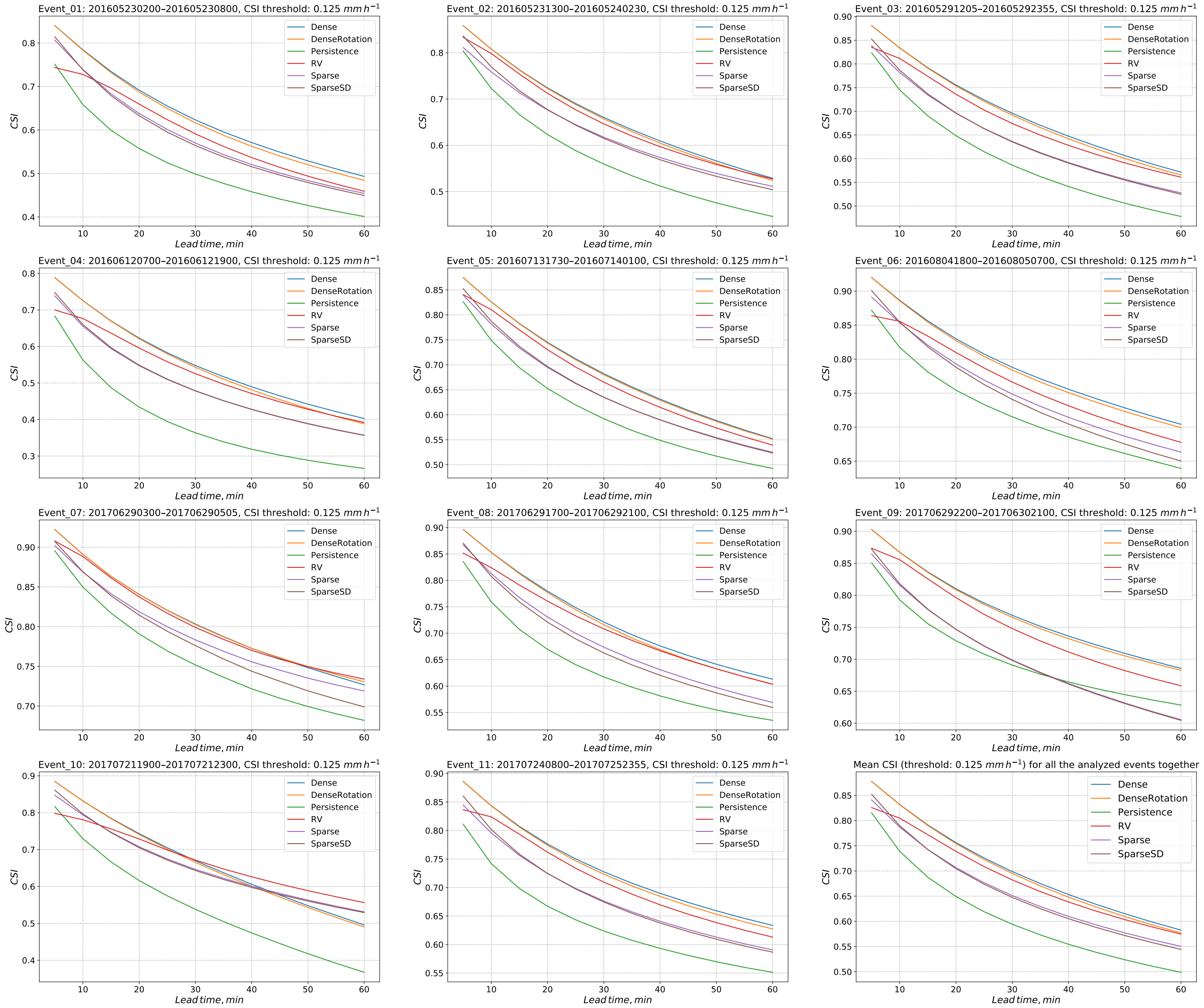

Figure 6. Lead time wise CSI for the threshold of 0.125 mm h^{-1}

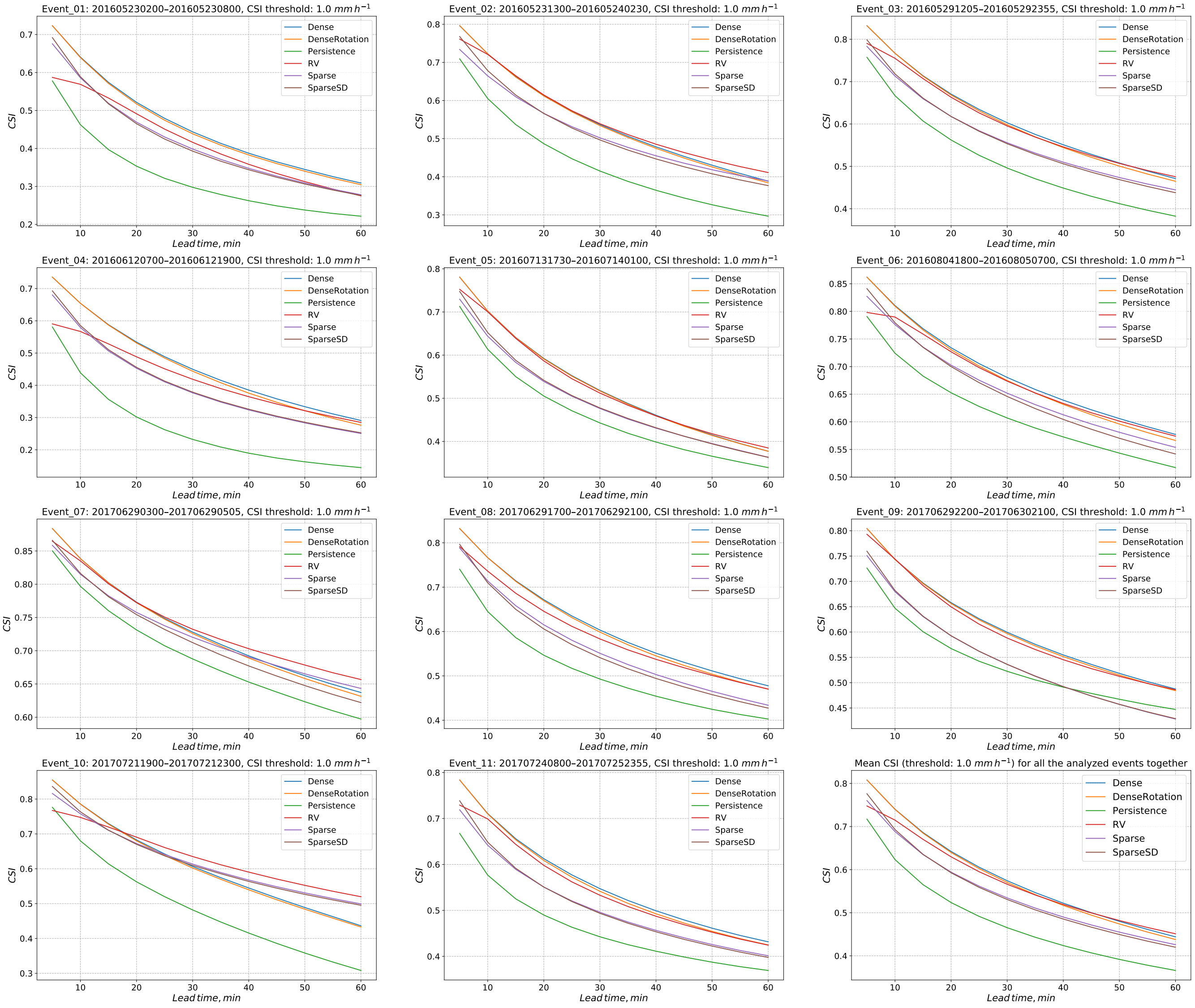

Figure 7. Lead time wise CSI for the threshold of 1.0 mm h^{-1}

Table 1. Overview of the developed models

Model name	Input radar images	Tracking	Extrapolation	Number of parameters	Computational time, s
SparseSD	2	Shi–Tomasi corner detector, Lucas–Kanade optical flow (local)	Constant delta-change, affine warping	7	~2-3
Sparse	3-24	Shi–Tomasi corner detector, Lucas–Kanade optical flow (local)	Linear regression, affine warping	10	~2-3
Dense	2	Farnebäck optical flow (global)	Constant-vector advection scheme	7	~150-180
DenseRotation	2	Farnebäck optical flow (global)	Forward semi-Lagrangian advection scheme	7	~150-180

Table 2. Characteristics of the selected events

Event shortcut	Start	End	Duration, hours	Maximum extent, km^2	Extent >1 mm h^{-1}, %
Event 1	2016-05-23 2:00	2016-05-23 8:00	6	159318	42
Event 2	2016-05-23 13:00	2016-05-24 2:30	13.5	135272	56
Event 3	2016-05-29 12:05	2016-05-29 23:55	12	160095	72
Event 4	2016-06-12 7:00	2016-06-12 19:00	12	150416	53
Event 5	2016-07-13 17:30	2016-07-14 1:00	7.5	145501	62
Event 6	2016-08-04 18:00	2016-08-05 7:00	13	168407	74
Event 7	2017-06-29 3:00	2017-06-29 5:05	2	140021	70
Event 8	2017-06-29 17:00	2017-06-29 21:00	4	182561	60
Event 9	2017-06-29 22:00	2017-06-30 21:00	23	160822	75
Event 10	2017-07-21 19:00	2017-07-21 23:00	4	63698	77
Event 11	2017-07-24 8:00	2017-07-25 23:55	16	253666	63

Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2018-166
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 10 September 2018

Table 3. Mean model metrics for different lead time periods

	Lead time (from–to), min	
Model	5–30	35–60
	MAE, mm h^{-1}	
Dense	**0.29**	**0.44**
DenseRotation	**0.29**	**0.44**
RV	0.31	0.45
	CSI, threshold=0.125 mm h^{-1}	
Dense	**0.78**	**0.63**
DenseRotation	**0.78**	0.62
RV	0.76	0.61
	CSI, threshold=0.25 mm h^{-1}	
Dense	**0.76**	**0.60**
DenseRotation	**0.76**	0.59
RV	0.74	0.59
	CSI, threshold=0.5 mm h^{-1}	
Dense	**0.73**	0.55
DenseRotation	0.72	0.55
RV	0.70	0.55
	CSI, threshold=1 mm h^{-1}	
Dense	**0.68**	0.49
DenseRotation	0.67	0.49
RV	0.65	0.49

The best result for specific metric and lead time period in **bold face**