

Identification of summer hailstorm from MSG data using Python

A.M. GUERRERO-HIGUERAS, A. MERINO, L. LÓPEZ AND J.L. SÁNCHEZ

Atmospheric Physics Group, University of León, León, Spain

V. MATELLÁN

Dpt. Mechanical, IT, and Aerospace Engineering, University of León, León, Spain

ABSTRACT

Severe storms with hail precipitation are one of the most common meteorological risks in Europe. The detection of these episodes is a habitual problem for organisms responsible for managing meteorological risks. The Group for Atmospheric Physics (GFA), from the University of León (ULE), has developed an algorithm capable of identifying this type of episode using information registered by a Meteosat Second Generation (MSG) satellite. Using this algorithm, the PyHTD tool allows to identify geographic areas with high probability of hail fall. This paper details how the PyHTD tool was constructed using the algorithm defined by the GFA.

1. Introduction

Severe storms with hail precipitation are one of the most common meteorological risks in Europe. They can provoke grave economic losses and even affect the population. Identifying these types of episodes is one of the most habitual demands of the population in general, and specifically, for risk management organisms. The Atmospheric Physics Group (GFA) from the University of León (ULE), has developed a model that allows for identification of hailstorms in real-time (nowcasting), using information gathered from a Meteosat Second Generation satellite (MSG).

The new generation of geostationary MSG satellites includes sensors that have a spatial-temporal and spectral resolution that is better than that of its predecessors. One of these sensors is a radiometer named Spinning Enhanced Visible and InfraRed Imager (SEVIRI). This sensor has twelve spectral channels with a sampling distance of 3 km and also has a high-resolution visible (HRV) channel with a sampling distance of 1km. The temporal resolution of SEVIRI is 15 minutes. These characteristics translate to a substantial improvement in following meteorological phenomenon with fast rates of growth and a relatively short lifetime, such as, for example, flash-floods, or convective storms, which makes MSG an extremely useful tool in following this type of phenomenon.

The information gathered by the SEVIRI radiometer from the MSG satellite presents a format distributed by EUMETSAT (European Organisation for the Exploitation of Meteorological Satellites) named HRIT. There are tools that transform the information in an HRIT format to other

more friendly formats, such as netCDF, which can also be treated using Python.

Calculating hail probability in real-time using the algorithm proposed by the GFA requires various steps. The first step is the definition of the geographical area of study. It is necessary to discretize this area in order to study it. In order to do so, a two-dimensional matrix to cover it is defined. We refer to this discretized model as a grid. Each point in the grid is called a pixel. The larger the number of pixels in the grid, the greater the resolution of the results obtained will be. Additionally, the computation time will also be greater.

The next step consists of calculating hail probability in each pixel. In order to do so, a multinomial logistic regression model is applied in two phases, using the step-wise method for incorporating variables. In the first phase, the clouds of interest to the study are separated from the rest of the cloud formations. Specifically, well-developed convective formations are detected, which are known as cumulonimbus clouds. In the second phase, the probability that hail precipitation will exist within these clouds is calculated.

Lastly, it is necessary to make a series of corrections to the results obtained in the previous step. The first correction consists of adjusting the probability of each pixel, taking into account its solar zenith angle, since the algorithm has been optimized for the calculating probability in day-time conditions.

Later, it is necessary to correct the coordinates of each pixel due to the Parallax effect. This distortion in the coordinates is produced because of the position of the satellite

in relation to the curvature of the Earth. The corrections are carried out according to the method proposed in Vicente et al. (2002).

Finally, following all of the above mentioned steps, the probability of hail precipitation for each pixel on the grid for the defined study area is obtained.

The resulting information can be directly represented graphically using Python. Additionally, the results can be exported to other formats, such as, for example, netCDF, readable by any tool that allows for the analysis and representation of this information: GrADS, NCL, Matlab, R, ncview, etc.

In summary, the methodology presented allows to construct graphic representations in which, for a chosen area of study, it is possible to know the probability that hail precipitation is being produced in real-time (nowcasting).

2. The hail detection tool: physical results

The GFA has developed a new methodology to identify hail fall events using Meteosat satellite data, Merino (2013). First a training database with several cloud type was established, including satellite and radar imagery from the summer months between 2006 and 2010 during day time hours.

Later, the algorithms were constructed using logistic regression by means of the stepwise method, where the MSG spectral channel were added according to their forecasting potential. The hail detection tool consists of two distinct stages. The first part includes a convective mask to identify cumulonimbus clouds, and the second part includes a hail mask to identify hail falls in cumulonimbus clouds.

The hail detection tool provides the probability of hail P obtained by the model and is expressed as follows:

$$P(X_1, X_2 \dots X_k) = p_1 = \frac{\exp(Z_1)}{1 + \exp(Z_1)} \quad (1)$$

where,

$$Z_1 = \alpha + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k;$$

X_j for $j = 1, \dots, k$, are the metric explanatory variables;

k is the total number of variables or interactions between variables included in the model;

α is a constant;

β_j for $j = 1, \dots, k$, are the various discriminating weights;

To create the convective mask the model carried out 10 iterations introducing 9 variables in the equation (Table 1), and for the hail mask the model carried out 8 iterations introducing 4 variables in the equation (Table 2). For the both models, the assessment of the global fit was carried out by means of the chi-square test used to assess the reduction of the -2LL parameter presents significant results (Sig <0.05), revealing a good fit of the model.

TABLE 1. Parameters selected by the logistic regression for the convective mask.

Metric explanatory variables	β
Intercept	1492.636
Brightness temperature 8.7 μm	1.188
Brightness temperature 6.2 μm	-5.186
Albedo 1.6 μm	2.226
Albedo 0.8 μm	-1.659
Albedo 3.9 μm	-0.884
Brightness temperature 7.3 μm	-7.627
Albedo 1.6 μm *	-0.009810
Brightness temperature 8.7 μm	
Brightness temperature 6.2 μm *	0.026309
Brightness temperature 7.3 μm	
Albedo 0.8 μm *	
Albedo 3.9 μm	0.007047

TABLE 2. Parameters selected by the logistic regression for the hail mask.

Metric explanatory variables	β
Intercept	115.039
Brightness temperature 6.2 μm	-0.624
Albedo 1.6 μm	-2.18
Albedo 0.8 μm	0.118
Albedo 1.6 μm *	
Brightness temperature 6.2 μm	0.010955

The preliminary application of the convective mask is crucial to filter the cloud tops formed by liquid water. The reason is that the hail mask is highly sensitive to the values of the near-infrared channels. Cloud tops formed by liquid water have high reflectances in these channels, so that they would eventually be classified as hail.

The hail detection tool validated using independent data from 2011 shows a probability of detection of 76.9% and a false alarm ratio of 16.7%. Although the performances are slightly worse than those characterizing the hail detection from ground-based radars, the hail detection tool is recommended itself for an application to areas not necessarily covered by radar systems and of any spatial extent. The hail detection tool will be used for the tracking and nowcasting of hailstorms in real time, using radar data to implement the system.

Nevertheless, the combination of the results provided by the hail detection tool with another ground true database requires a correction for the Parallax effect. In the case of the MSG satellite, the parallax effect is important because the satellite is over the equator, taking measurements of Europe at relatively low angles. Lábó et al. (2007) found a deviation towards the southwest of up to four pixels for

high clouds over Hungary. The Iberian Peninsula is not situated at the satellite's nadir and the analyzed cumulonimbus clouds have high tops and thus this correction must be applied. The deviations computed in this study follow the methodology by Vicente et al. (2002), which takes data of cloud top height from the vertical reflectivity profiles obtained by the radar. For example, for high clouds the deviations found were of up to 18 km to the south, and very small deviations were found in the E-W direction because the meridian 0 crosses the study area. Eventually, this correction enables us to compare satellite data and radar data for the same point of the surface.

3. Software Architecture

The GFA has developed a tool, using Python and NCL, called PyHDT¹, that implements a calculation of hail probability, as shown in the previous section. PyHDT is composed of two scripts:

- **hail.py**: Calculate the convection mask and the hail mask. Additionally, correct the values of probability at each point, at the base of the zenith angle. The source code for **hail.py** is available on line at GFA website².
- **hail.ncl**: Generates a graphic that allows us to visualize the probability of hail. Also, it corrects for the Parallax effect in the MSG satellite coordinates. The source code of **hail.ncl** is available on line at GFA website³.

a. Calculation of hail probability

As explained previously, the probability of hail is calculated in two steps. In the first step, a convection mask is generated. In the second, a hail mask is calculated for each of the points of the convection mask. In both steps, Equation 1 is applied with the metric explanatory variables in Tables 1 and 2, respectively.

In order to obtain the values of the metric explanatory variables, the GFA has used information given from the *SatDatos* application, PRAPRO S.L. (2009). *SatDatos* transforms the information received by the MSG receptor from an *HRIT* format to a *netCDF* format. *SatDatos* produces a *netCDF* file that includes a variable, **britemp**, with the information about all of the metric explanatory variables that are necessary, as seen in Table 3.

¹Copyright 2013 Atmospheric Physics Group, University of León, Spain. PyHDT is distributed under the terms of the GNU Lesser General Public License.

²<http://gfa.unileon.es/data/toolbox/PyHDT/hail.py>

³<http://gfa.unileon.es/data/toolbox/PyHDT/hail.ncl>

TABLE 3. Metric explanatory variables obtained from *SatDatos*.

Metric explanatory variables	
Albedo 0.6 μm	C_1
Albedo 0.8 μm	C_2
Albedo 1.6 μm	C_3
Albedo 3.9 μm	C_4
Brightness temperature 6.2 μm	C_5
Brightness temperature 7.3 μm	C_6
Brightness temperature 8.7 μm	C_7
Brightness temperature 9.7 μm	C_8
Brightness temperature 10.8 μm	C_9
Brightness temperature 12.0 μm	C_{10}
Brightness temperature 13.4 μm	C_{11}
High resolution visible	C_{12}

1) CALCULATION OF THE CONVECTION MASK

In order to calculate the convection mask, P_0 , Equation 1 is applied. We multiplied the result by 100 in order to have probability values between 0 and 100, as shown in Equation 2.

$$P_0 = 100 * \left(\frac{\exp X}{1 + \exp X} \right) \quad (2)$$

where X is calculated by applying the metric explanatory variables and the β coefficients of Table 1, as shown in Equation 3.

$$\begin{aligned} X = & 1492.636 \\ & + (1.188 * C_7) \\ & + (-5.186 * C_5) \\ & + (2.266 * C_3) \\ & + (-1.659 * C_2) \\ & + (-0.884 * C_4) \\ & + (-7.627 * C_6) \\ & + (-0.00980977 * C_3 * C_7) \\ & + (0.02630949 * C_5 * C_6) \\ & + (0.00704733 * C_2 * C_4) \end{aligned} \quad (3)$$

The calculation of the convection mask for all the points on the grid is done with the **getConvectiveMask** function of the **hail.py** script.

2) CALCULATION OF HAIL MASK

In order to calculate the hail mask, P_1 , we also applied Equation 1. We multiplied the result by 100, in order to have probability values between 0 and 100, and for the convection mask, P_0 , in order to work only with the points that stand out, as seen in Equation 4.

$$P_1 = P_0 * 100 * \left(\frac{\exp Y}{1 + \exp Y} \right) \quad (4)$$

Prior to doing this, it is necessary to discretize the values of the convection mask, P_0 , to indicate the points on the grid that produce hail. To do this, we assigned a value of 1 to the points with a probability greater than 50 and a value of 0 to the points with a probability lower than 50.

Y is calculated by applying the metric explanatory variables and the β coefficients of Table 2, as seen in Equation 5.

$$\begin{aligned} Y = & 115.039 \\ & + (-0.624 * c_5) \\ & + (-2.180 * c_3) \\ & + (0.118 * c_2) \\ & + (0.01095546 * c_3 * c_5) \end{aligned} \quad (5)$$

The `getHailMask` function of the `hail.py` script calculates the hail mask for all the points on the grid.

b. Calculation of the zenith Angle

In order to correct the probability values for a given instant, it is necessary to know the values of the zenith angle, θ , for each point at that instant. In order to calculate the value of θ , at a determined instant and point, we should follow these steps:

- i. The value of δ is calculated according to Equation 6:

$$\delta = 23.45 * \sin \left(360 * \left(\frac{284 + J}{365} \right) \right) \quad (6)$$

where J corresponds to the day of the year.

- ii. The solar hour, H , is calculated using Equation 7:

$$H = hour + \left(\frac{4 * lon * 100}{60 * 100} \right) \quad (7)$$

where $hour$ is the hour UTC for the instant that we are calculating the zenith angle and lon is the value of longitude, in degrees, from the point where we are calculating the zenith angle.

- iii. The value of ω is calculated using Equation 8:

$$\omega = 15 * (H - 12) \quad (8)$$

- iv. The value of the zenith angle, θ , is calculated using Equation 9:

$$\theta = \arccos(\sin \delta * \sin \phi + \cos \delta * \cos \phi * \cos \omega) \quad (9)$$

where ϕ corresponds to the value of latitude, in radians, of the point from which we are calculating the zenith angle.

The `zenithAngle` function of the `hail.py` script calculates the zenith angle for all the points of the grid.

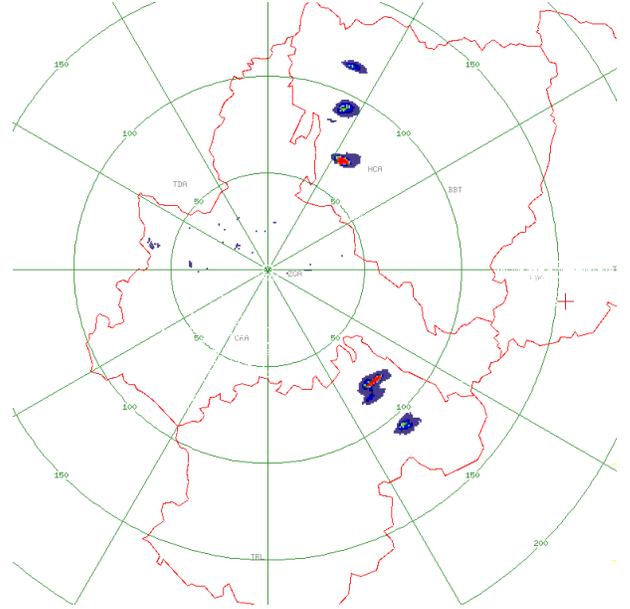


FIG. 1. Radar image at 16:00 UTC on 21 July 2010.

c. Parallax correction

The correction for the Parallax effect, in the MSG satellite coordinates, has been carried out according to Vicente et al. (2002). The correction is done using the `parallax` function of the `hail.ncl` script. In order to make the correction, it is necessary to know the height of the cloud at each point. To do so, we have compared the value of the *brightness temperature 10.8 μm* variable, with a vertical temperature profile obtained from the output given by the WRF model, which the GFA executes operationally. The `cloudHGT` function of the `hail.ncl` script performs this calculation.

4. Development Process

PyHDT, Copyright 2013 Atmospheric Physics Group, was developed using *NumPy* and *SciPy* and libraries, Jones et al. (2001). The scripts are available to the scientific community under the terms of the GNU Lesser General Public License^{4,5} (LGPL). They are available on line at GFA website⁶.

5. Application: 2010 July 21th episode

In order to demonstrate PyHDT's performance, we have selected an episode which shows data that allows to corroborate its performance. The episode analyzed corresponds to 21 July 2010, a day when important convective

⁴<http://www.gnu.org/licenses/gpl.txt>

⁵<http://www.gnu.org/licenses/lgpl.txt>

⁶<http://gfa.unileon.es/?q=toolbox>

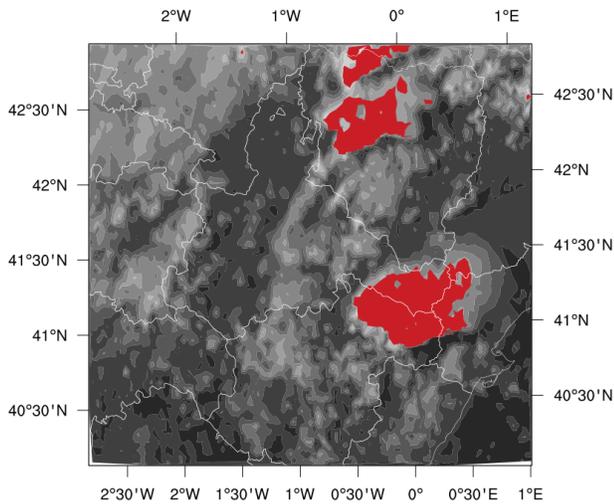


FIG. 2. Convection mask at 16:00 UTC on 21 July 2010.

hailstorms were registered in the northeast quadrant of the Iberian Peninsula. Thus, in the Aragon region of Bajo Aragon, various hailstorms were registered at approximately 16:00 UTC, causing significant damage to crops. In Figure 1, the regions with an elevated probability of being affected by hail are shown in red, according to the information registered by the GFA’s meteorological radar.

Figure 2 shows the results obtained from applying the convective mask in the case of study, over the northeastern quadrant of the Iberian Peninsula. The clouds determined by the algorithm to be cumulonimbus are shown in red. Thus, we can appreciate diverse convective nuclei in the study area, which are correctly filtered from other types of clouds present.

At this same instant, in Figure 3, we can see the results, on a probability scale, of applying the hail mask over the convection mask. We can see elevated probability of hail in the regions within the cumulonimbus clouds.

6. Conclusion

If we compare Figures 3 and 1, we can see that in both cases, very similar results were obtained. This demonstrates that PyHDT is a useful tool for detecting convection in the form of hail in real-time.

The results provided by the tool are useful for registering hail fall events in daytime hours during summer months. This tool complements meteorological radar in the regions where the measurements provided by the radar are not precise. Meteorological radar suffers important signal reductions when strong precipitation is present. In addition to mountainous areas, such as the Iberian Peninsula, the radar’s range is lower, and it is necessary to have a dense radar network available to cover the entire territory,

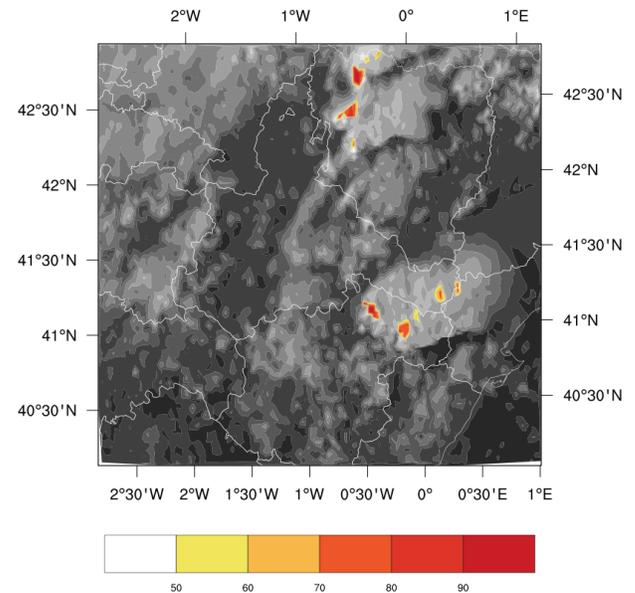


FIG. 3. Hail probability at 16:00 UTC on 21 July 2010.

which notably increases the cost. However, this tool has an increased spatial coverage and lower economic costs.

Acknowledgments.

The authors would like to thank the Junta of Castile and León for their economic support via the LE220A11-2 project. This study was supported by the following grants: CGL2010-15930; MICROMETEO (IPT-310000-2010-22).

REFERENCES

- Jones, E., T. Oliphant, P. Peterson, et al., 2001: SciPy: Open source scientific tools for Python. <http://www.scipy.org>.
- Lábó, E., J. Kerényi, and M. Putsa, 2007: The parallax correction of MSG images on the basis of the SAFNWC cloud top height product, Proc. *EUMETSAT Meteorological Satellite Conf. and 15th Satellite Meteorology & Oceanography Conf. Amer. Meteor. Soc.*, Amsterdam, The Netherlands.
- Merino, A., 2013: Analysis, identification and forecast of hail precipitation events in the iberian peninsula. Ph.D. thesis, University of León, 180 pp.
- PRAPRO S.L., 2009: Nubes software. www.prapro.com.
- Vicente, G., J. Davenport, and R. Scofield, 2002: The role of orographic and parallax corrections on real time hight

resolution satellite rainfall rate distribution. *Int. J. Remote Sens.*, **203**, 221–330.