19th Symposium IMEKO TC 4 Symposium and 17th IWADC Workshop
Advances in Instrumentation and Sensors Interoperability
July 18-19, 2013, Barcelona, Spain

CLOUD NOWCASTING: MOTION ANALYSIS OF ALL-SKY IMAGES USING VELOCITY FIELDS

Yézer González[1], César López[1], Emilio Cuevas[2]

[1] Sieltec Canarias S.L. (Santa Cruz de Tenerife, Canary Islands, Spain. Tel. +34-922356013, Fax +34-901706711. E-mail: yezer.gonzalez@sieltec.es)
[2] Izana Atmospheric Research Center (AEMET, Spain)

Abstract
Cloud monitoring and prediction is crucial to air traffic and weather forecast. Clouds play a key role in solar radiation balance. These issues have prompted Sieltec Canarias S.L. and AEMET to develop an automatic system of cloud observation (SONA). This system provides cloud cover percentage using all-sky images, which are processed by neural networks. Once the system has detected and recorded cloud images, a cloud motion technique makes possible to obtain "velocity fields", corresponding to different layers of clouds. The evolution in time of these velocity fields is a powerful tool to provide very short term forecasting (less than 1 hour) of cloudiness (known as cloud nowcasting). In the future, this information, combined with Radiative Transfer Models, may be used for solar radiation nowcasting (Global Horizontal Irradiance –GHI- and Direct Normal Irradiance –DNI-)

I. INTRODUCTION

Cloudiness or cloud fraction is a meteorological observation traditionally performed by observers. They look at the sky and estimate the cloud cover in oktas. These observations are expensive and are not accurate enough because of human subjectivity. We have developed SONA (Automatic Cloud Observation System). This system provides, with high spatial and temporal resolution, hemispherical images of the sky, which are real time processed and analysed providing an objective and accurate full map of cloudiness. One-minute animated frames are used to detect the clouds motion and forecast their evolution.

II. INSTRUMENTATION

SONA comprises an All-Sky imager with the following hardware: a camera with a CCD sensor with Bayer filter and resolution 640x480 pixels, 8 bit color response from 400 to 700nm, and monochrome response from 400 to 1000nm. This camera is inside a very durable aluminium housing. The CCD with a fish-eye lens points to the zenith through a borosilicate dome. The system incorporates a rotating shadow band for protecting the sensor from direct sunlight. The control electronics is made from industrial-grade components and safe against lightning. The operating temperature is from -10º to +50ºC thanks to its cooling/heating system. SONA can also include a simple multichannel photometer to determine the atmospheric aerosols content and modify the behaviour of the cloud detection algorithms accordingly. SONA has been developed and tested at the Izaña Atmospheric Observatory (IZO; 18.5ºN 16ºW; 2,400 m a.s.l.), managed by AEMET (Meteorological State Agency of Spain).

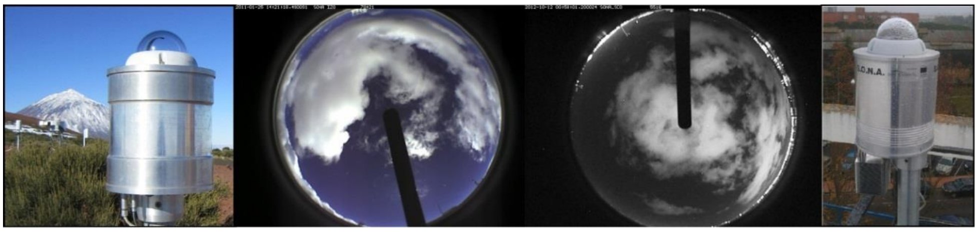

Figure 1. Left: The SONA V3i, and its aluminium housing, dome and rotating shadow band. Center: Sky image taken by SONA during daytime and night-time, respectively. Right: SONA V6.

ISBN-10: 84-616-5438-2 | ISBN-13: 978-84-616-5438-3

19th Symposium IMEKO TC 4 Symposium and 17th IWADC Workshop
Advances in Instrumentation and Sensors Interoperability
July 18-19, 2013, Barcelona, Spain

III. PROCEDURE

Cloud cover characterization with sky images is a complex problem. Many variables related to the sky such as luminosity, sun position, cloud distribution, types of clouds, atmospheric aerosol/dust content, etc., as well as factors related with the hardware, as iris aperture, exposure time, contrast, color gain, geometrical distortion and noise, introduce a great degree of complexity in automatic cloud monitoring.
This complexity requires an approach in which the most essential features and corresponding multiple interactions might be modelled. Neural networks are flexible, adaptive learning systems that find patterns in data of a nonlinear system as cloudiness. We use a multilayer perceptron network (MLP) for image pattern recognition (Fig. 2C). MLP is widely used nonlinear network for solving many practical problems in applied sciences for data classification MLP can be trained as classifier as follows: and input pattern (x_1, x_2, ..., x_n) is transmitted through input connections (one input connection for one input data) whose inputs weights (I_{11}, I_{12}, ..., I_{1n}, ..., I_{p1}, I_{p2}, …, I_{pn}) are initially set to random values. Those inputs are summed and processed in every neuron of the first layer. The outputs of the first layer are sent to the second layer of neurons with their initial random weights (a_{11}, a_{12}, a_{13}, ..., a_{1n}, ..., a_{p1}, a_{p2}, ..., a_{pn}). In this way the information is propagated by the whole network. When the process arrives to the last neuron layer, the final outputs are generated and compared with the given target output to determine the error (E) for this pattern. Inputs and patterns are presented iteratively and the weights are adjusted until the minimum possible square error (MSE) when the entire training dataset have passed through the network. MSE is calculated using a back-propagation method [2]. Training dataset must be launched in the network many times for better results.

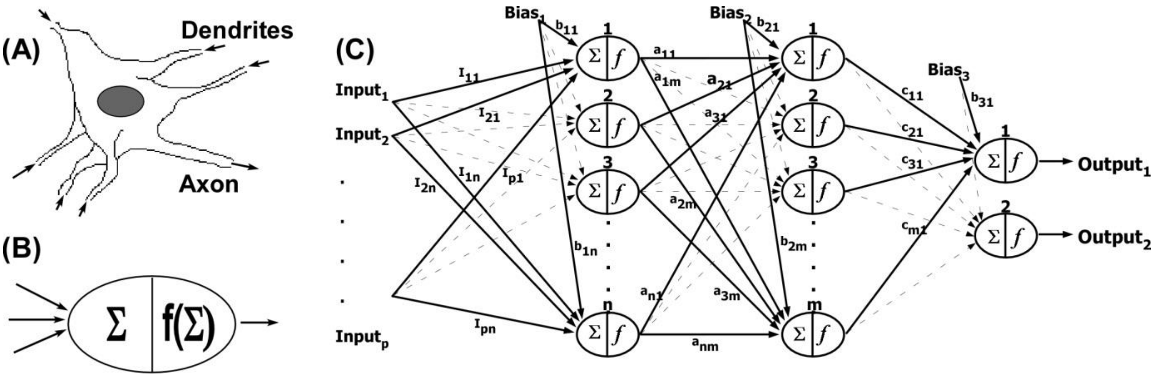

Figure 2. (A) Biological network. (B) Computer neuron model or perceptron. (C) Multilayer neural network with an input layer, two hidden neuron layers and a neuron output layer.

Before the cloud cover detection it is necessary to choose a set of representative images from sky, which are the input for neural network training algorithm. Firstly, the algorithm places a virtual mask in the original image over the shadow band and the obstacles observed in the outer margins of the image. So, we clean the images, keeping only pixels corresponding to the sky. Then we select the target, cloud or clear pixels, and this information is saved with all pixel input parameters.
The inputs parameters are related to every single pixel and its neighbourhoods in the sky image. The inputs parameters are the following:
- Red, green, blue values (0-255) and their variance values. These values are calculated in the 8-neighborhood pixels.
- Distance from Sun to pixel, zenith to pixel and Sun to zenith (pixels).
- Hardware factors.

Configuring and training the MLP reaching the minimum possible square error (discussed above) is the last step. The algorithm saves the configuration of the trained neural network in “memory parameters”, which are stored in a text file.
The algorithm of cloud cover detection in sky images starts obtaining a sky image (Fig. 3A). Later the position of the shadow band is detected placing a similar mask over it (Fig. 3B) in order to eliminate highlights and border effects in the image, because the less pixels there are, the less time computing. The algorithm analyses every single pixel and its 8-neighboorhood pixels of the image calculating their parameters (Fig. 3C). These parameters feed the trained neural network (Fig. 3D), that way neural network determines whether the pixel corresponds to cloud or to clear sky. Finally, the algorithm generates an output image with detected cloudy pixels (Fig. 3E) and a percentage of cloudiness.

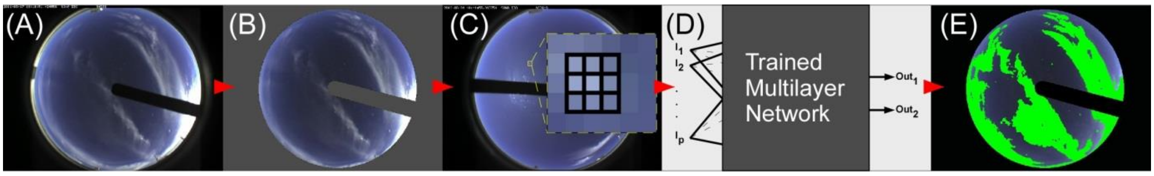

Figure 3. Cloud detection steps. (A) Original image. (B) Masked image. (C) Algorithm analyses sky pixels and extracts input parameters. (D) MLP generates the type of the pixel. (E) Output image with cloud pixels marked.

Some environmental and hardware constraints might introduce wrong cloudiness estimation. For example, equal number of pixels in the image might correspond to different solid angles [3]. A geometric correction must be then applied to every single pixel. Identification of cloudiness in nearby images in time allows us to improve the optical flow calculation between consecutive images using Farneback´s algorithm [4] (Fig. 4). The goal of optical flow estimation is to compute an approximation to the motion field from time-varying image intensity. Some different time scales to calculate flow motion is needed, due to the fact that apparent motion of high clouds is less than that of lower clouds.

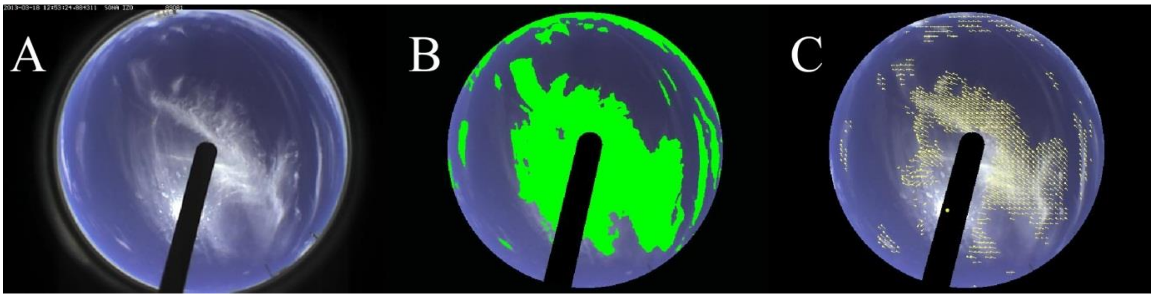

Figure 4. (A) Original image. (B) Image analysed by the neural network with cloud pixels marked as green. (C) Image with the whole velocity field marked as yellow.

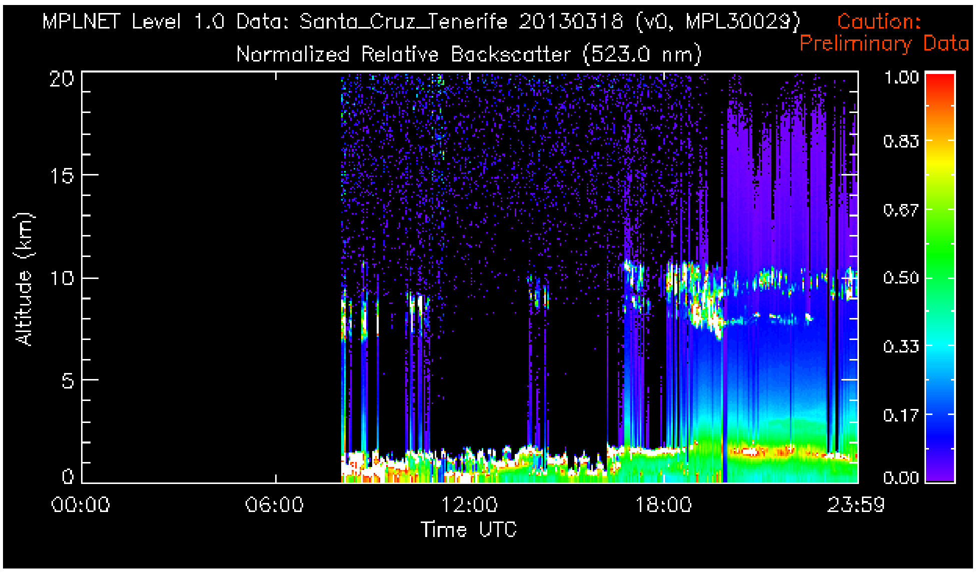

Figure 5. Normalized relative backscatter vertical cross section obtained with the Micropulse Lidar at Santa Cruz de Tenerife on March 18th, 2013 (MPLNET).

As an example we have validated the motion of clouds shown in Figure 4 on March 18th, 2013. The Micropulse lidar, is co-located to the SONA camera. The lidar detected the cirrus clouds observed by SONA at around 9 km height over Santa Cruz de Tenerife. The meteorological radiosonde reported a wind direction of 250º with wind speeds of 40m/s between 8 and 10 km altitude. According to the optical flow calculation, the wind direction associated to cirrus movement is 243º, in good agreement with radiosonde measurements.

Once optical flow has been calculated, it is necessary to cluster the whole motion field in the image into subfields which are formed by motion vectors with similar features. So, we can spot clouds with different relative motion. This relative motion might be due to the fish eye lens of the camera, or to the presence of different altitude cloud layers.

To cluster the motion field we have based on a Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise (DBSCAN) [5]. DBSCAN allows to discovery clusters with arbitrary shape, providing good efficiency in large databases. We have developed a modified version of DBSCAN. In our version we group the vectors in a different way avoiding vectors with module value equal to zero, so the algorithm does not evaluate all the points, and the processing time is shorter.

Our algorithm groups motion vectors with similar direction and module (Fig. 6A), detecting the noise, which usually are few error vectors result of Farneback´s algorithm. They arise in the borders of blocking objects in the image as the rotating shadow band or the Earth horizon. In case this noise is present, it is removed. Every detected cluster gives an average vector (Fig. 6B). This is important in order to reduce computational time and for predicting cluster position in next developments.

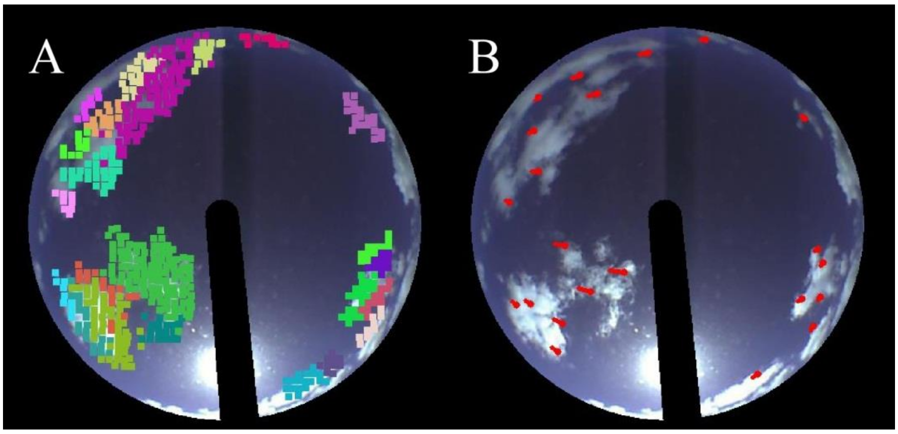

Figure 6. (A) Clustered image with modified DBSCAN algorithm Clusters are marked with different random colours. (B) Image with average motion vectors (in red color) corresponding to clusters.

Using average vectors we can detect wind direction. Specifically, we analyse the vectors whose directions are closer to the zenith, following a near straight line which gives us an approximate wind direction. The system saves the average vectors of some few consecutive previous images. This way we obtain an accurate wind direction associated to clouds movement (Fig 7.A).

Taking into account the wind direction, the algorithm calculates the trajectories of every average motion vector and therefore, the orbits of the clusters belonging to clouds. The trajectories are fitted to functions depending on the distance from the average vector to zenith in the image (Fig. 7B).

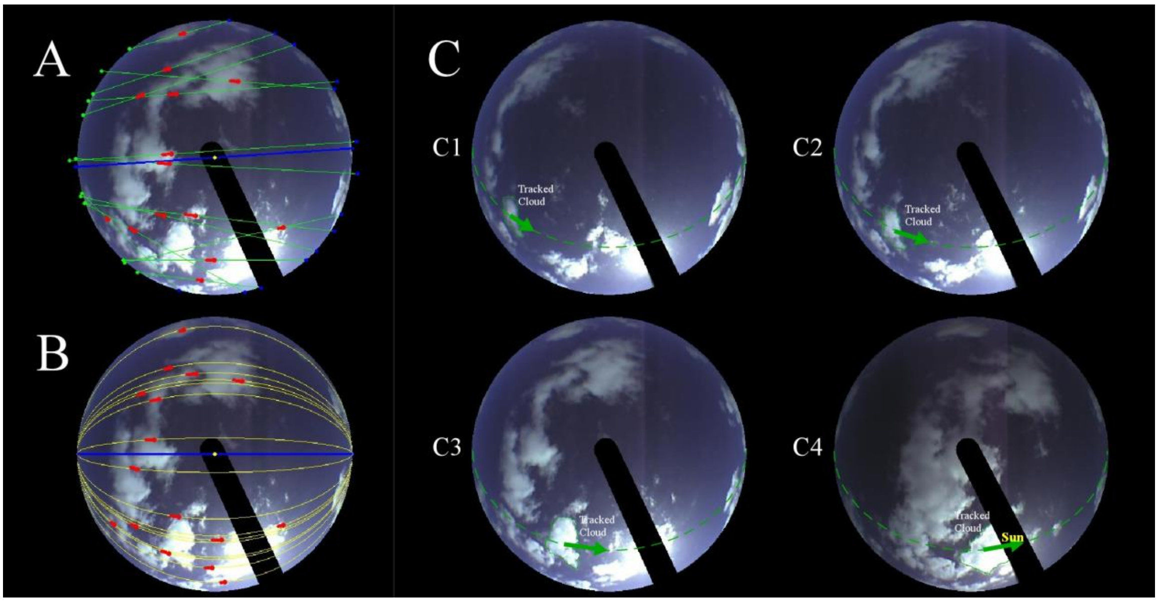

Figure 7. (A) Calculation of wind direction (blue line) based on average cluster vectors. (B) Trajectories corresponding to the cluster vectors. Every vector follows a calculated orbit function. (C) Depending on future evolution of each vector in its own trajectory it is possible to estimate whether the Sun will be blocked by clouds.

The future position of a cluster can be predicted if the trajectory function, and its corresponding average vector, are known. So, we can determine the future vector position in its orbit. This technique allows us to predict when the Sun will be blocked by clouds, what is quite interesting for solar energy applications (Fig. 7C). In Figure 7 we have a good example of cloud tracking. The selected cloud moved from position marked in C1 to position marked in C4 in only 195 seconds. This is the worst case, in which the Sun is very low over the horizon, the clouds are low and moving at high speed, so that the time between the detection of the cloud until it blocks the sun is very short. Previous works obtain a global vector for the behaviour of clouds scene, but they do not take into account the different cloud layers with different morphology, height and velocity vectors, and the field distortion due to the fish eye optics.

IV. CONCLUSIONS

We have developed and implemented a very efficient system for cloud detection. In combination with our "velocity-fields" clustering algorithm, our software is able to provide cloud nowcasting. This is a necessary first step for solar radiation nowcasting.

V. REFERENCES

1. Bishop, C.M., *Neural Networks for Pattern Recognition*. 1996: Oxford University Press, USA.
2. Samarasinghe S., *Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition*. 2007: Auerbach.
3. Long, C.N., Sabburg, J. M., Calbó, J., Pagés, D., *Retieving cloud characteristics from ground-based daytime color all-sky images.* J. Atmosf. Oceanic Technol., 2006. **23**: p. 633-652.
4. Farneback, G. *Two-frame motion estimation based on polynomial expansion*. in *Proceedings of the 13th Scandinavian conference on Image analysis*. 2003. Halmstad, Sweden: Springer-Verlag.
5. Ester, M., et al., *A density-based algorithm for discovering clusters in large spatial databases with noise.* 1996: p. 226-231.