A NEW CLOUD CLASSIFICATION SYSTEM FOR RAINFALL DETECTION OVER PUERTO RICO USING REMOTELY SENSED DATA

by

Melvin J. Cardona-Soto

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
COMPUTER ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2011

Approved by:

________________________________ ________________
Hamed Parsiani, PhD Date
President, Graduate Committee

________________________________ ________________
Nazario D. Ramirez Beltran, PhD Date
Co-President, Graduate Committee

________________________________ ________________
Eric W. Harmsen, PhD Date
Member, Graduate Committee

________________________________ ________________
Sandra Cruz Pol, PhD Date
Member, Graduate Committee

________________________________ ________________
Victor Huérfano, PhD Date
Representative of Graduate Studies

________________________________ ________________
Erik E. Aponte, PhD Date
Chairperson of the Department

ABSTRACT

A new rainfall detection algorithm was developed to overcome challenges algorithms like the operational NOAA/NESDIS Hydro-Estimator (HE) present over Puerto Rico when detecting rainfall. The HE, a brightness temperature and numerical weather prediction based algorithm, detects about half of the rainfall received throughout a year, and when it does, the detection of rainfall is inconsistent. Part of this may due to the fact that the HE uses brightness temperature to discriminate between rain and no rain, and a large amount of the rainfall received in Puerto Rico is produced by warm clouds. In order to achieve greater accuracy of detection over PR, the new rainfall detection algorithm utilizes data from multiple channels of GOES-12 to extract several features from clouds (e.g., Brightness Temperature, Visible Reflectance, and Albedo). These features are utilized to perform a supervised classification of the image pixels into 4 previously defined classes. The classes were defined using NEXRAD rainfall detection information.

Radar and satellite information from five heavy storms that occurred from 2003 to 2007 were used to define the parameters of the algorithm. Once the algorithm was developed a discrete validation was performed to both the cloud classification system and the HE using NEXRAD information as ground truth. The performance of both algorithms was measured in terms of rainfall detection and compared. Warm cloud detection capability was also measured and compared between both algorithms.

RESUMEN

Un nuevo algoritmo de detección de lluvias se ha desarrollado para superar retos similares a los que presenta la versión operacional del Hidro-Estimador (HE) de NOAA/NESDIS sobre Puerto Rico al detectar precipitación. El HE, un algoritmo basado en temperatura de brillantez y datos de predicción numérica, detecta alrededor de la mitad de la precipitación recibida a lo largo de un año en Puerto Rico, y cuando lo hace, la misma es irregular. Parte de esto se debe a que el HE utiliza temperatura de brillantez para discriminar entre la lluvia y no lluvia, y una gran cantidad de la precipitación recibida en Puerto Rico es producto de nubes tibias. Con el fin de lograr una mayor precisión de la detección a través de PR, el nuevo algoritmo de detección utiliza datos provenientes de múltiples canales de GOES-12 para extraer diversas características de las nubes (ejemplo, temperatura de brillantez, reflectancia del canal visible y albedo). Estas características se utilizan para llevar a cabo una clasificación supervisada de los píxeles de la imagen en 4 clases definidas previamente utilizando información de NEXRAD.

Información de radar y satélite de cinco tormentas que se produjeron entre 2003 y 2007 se utilizaron para definir los parámetros del algoritmo. Una vez desarrollado el algoritmo se llevó a cabo una validación discreta con el sistema de clasificación de las nubes y el HE, utilizando información de NEXRAD como datos reales de campo. El rendimiento de ambos algoritmos se midió en cuanto a detección de lluvia y luego fueron comparados. La capacidad de detección de nubes tibias también fue medida y se compararon ambos algoritmos.

To everyone that in some way helped me achieve my goals…

ACKNOWLEDGEMENTS

During the development of my graduate studies in the University of Puerto Rico, Mayagüez Campus, several persons and institutions collaborated directly and indirectly with my research. Without their support it would have been impossible for me to finish my work. That is why I wish to dedicate this section to recognize their support.

I want to start expressing a sincere acknowledgement to Dr. Hamed Parsiani for the opportunity of researching under NOAA-CREST (National Oceanic and Atmospheric Administration - Cooperative Remote Sensing Science and Technology Center); Bob Kuligowski (NOAA/NESDIS), Daniel Lindsey (CIRA), Joan M. Castro and Santa E. Gomez who transmitted and offered their scientific and technical support. Special thanks I owe to Dr. Eric W. Harmsen and Dr. Nazario Ramirez-Beltran, Dr. Sandra Cruz Pol, because from them I received guidance, supervision, encouragement and support.

To NOAA-CREST, grant NA06OAR4810162, and Engineering Research Centers Program of the National Science Foundation, grant 0313747, which provided funding and resources for the development of this research.

Last, and most important, I would like to thank my family, for their unconditional support, inspiration and love towards me.

Table of Contents

Table List

Figure List

Figures **Page**

1 INTRODUCTION

1.1 Problem Statement

Rainfall detection and estimation algorithms based on brightness temperature extracted from remotely sensed data, such as the National Oceanographic and Atmospheric Administration's (NOAA) operational version of the Hydro Estimator (HE) (*Scofield and Kuligowski*, 2003), exhibits some problems when rainfall is estimated over Puerto Rico (PR). For instance, the Hydro Estimator, a brightness temperature and numerical weather prediction based algorithm, detects about half of the rainfall received throughout a year (*Harmsen et al., 2008*), and when it does, the detection rainfall is inconsistent. This may be due to the following reasons: 1) the algorithm was calibrated to work over the continental United States, 2) rainfall detection is based on a brightness temperature threshold, 3) a large amount of the rainfall is produced by warm clouds, 4) cloud features are extracted from a single satellite channel and weather prediction model files for correction, and 5) all of the rain detected is the product of a single relationship curve regardless of the type of the cloud. Figure 1.1 shows that at a given time almost half of the rainfall received are produced by warm clouds, and that the HE did not detect most of the rainfall detected by National Weather Service's (NWS) Next Generation Radar (NEXRAD) located in Cayey, PR since most of the rainfall was produced by warm clouds.

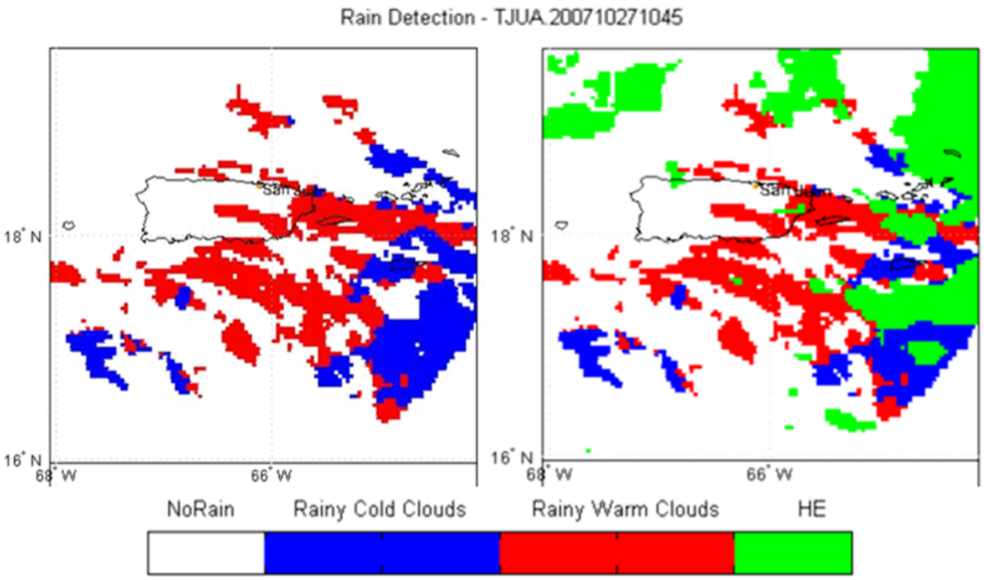


Figure 1.1 (Right side) Rainfall detected by NEXRAD. Pixels in blue represent rainfall produced by cold clouds, while red pixels represent rainfall produced by warm clouds (Right). (Left side) HE (green) warm cloud detection issue over Puerto Rico.

1.2 Justification

Being able to detect and estimate rainfall are the key element to protect human lives and infrastructure. Having this information on a timely manner can help to mitigate or prevent damage produced by floods. Remote sensing methods are important because they can provide valuable information over areas where rain gauges and other sources of hydrological information are scarce and only provide information for a given point. Information provided by these methods, especially radar and satellite data, may be used for flood forecasting, and numerous applications in the hydrometeorological sciences (*Vila and Velasco*, 2002) and by the agricultural community. With this motivation, researchers have developed several algorithms to detect and estimate rainfall utilizing satellite data.

The Hydro-Estimator (HE) algorithm (*Scofield and Kuligowski*, 2003), was designed to operate over the continental United States (U.S.) and satisfactory results have been reported.

1.3 Objectives

This study has various objectives; including 1) to create a cloud classification system to improve rainfall detection; 2) to identify remotely sensed variables that may improve detection of rainfall, especially rainfall produced by clouds with brightness temperature over 235K; 3) to develop a rainfall detection algorithm, which combines a cloud classification system and multiple remotely sensed variables obtained from geostationary satellite; 4) to validate the performance of the algorithm using NEXRAD data as ground truth.

1.4 Summary of the Following Chapters

Chapter 2 presents the literature review for this study. Various rainfall detection alternatives are described as well as five popular rainfall algorithms. This discussion is oriented to rainfall detection techniques implemented on the following algorithms: Hydro-Estimator, GMSRA, PERSIANN, PERSIANN-CCS and SCaMPR.

The third chapter deals with the methodology pertinent to this work; experiments and data analysis related to the rainfall detection improvements. Details are offered on the implemented supervised classification method as well as the steps taken to: a) identify the classes for rainfall detection, b) select the training data set for the supervised classification, c) develop the process to determine, which variables enhance rainfall detection over Puerto Rico, d) get remotely sensed products to be studied, and e) measure rainfall detection performance.

Results of every step mentioned on chapter 3 are presented in Chapter 4 as well as rainfall detection performance obtained by the developed algorithm. Finally in the fifth and final chapter Conclusions, Recommendations and Future Work are presented.

2 Literature Review

There exist several methods, instruments and sensors to detect and measure rainfall. Some of these are ground based (e.g., rain gauges and radars) while others are mounted over satellite platforms (e.g., imagers and sounders). Sections 2.1, 2.2 and 2.3 presents instruments used in this study to develop and validate the rainfall detection algorithm. Section 2.4 presents algorithm that in some way take advantage of the presented instruments.

2.1 Rain Gauges

Rain gauges are instruments that offer ground level measurements of precipitation. The presence of these has been recorded in history since the Korean Chosun Dynasty (1392–1910), and accredited to King Sejong who lived from 1397 to 1450 and was the fourth king of the Chosun Dynasty (*Jung*, 2001). All rain gauges were of the non-recording type until Sir. Christopher Wren (1632-1723) invented the first automatic tipping bucket. Unlike the previous versions of the rain gauges Wren’s invention recorded the measurements (*Biswas*, 1967). In order to measure rainfall every time the tipping bucket emptied it triggered a mechanism, which punched holes in a paper. The number of holes punched is proportional to the amount of rainfall. Automatic tipping bucket rain gauges are still being utilized for rainfall studies. Agencies such as the National Weather Service, the U.S. Geological Survey, the U.S. Army Corps of Engineers state districts, the National Park Service, and the U.S. Forrest Service have been using rain gauges for rainfall measurements and studies (*Habib*, 2008). Figure 2.1 presents an example of state of the art tipping bucket rain gauge. In Puerto Rico 125 rain gauges are distributed over the entire island, which collects rainfall measurements every 15 minutes (*USGS*, 2011). Rainfall data recorded by this rain gauge network is available since January 2000. Smaller rain gauge networks have also been installed in PR for rainfall related studies. In the following section an example of these small-scale rain gauge networks is described.

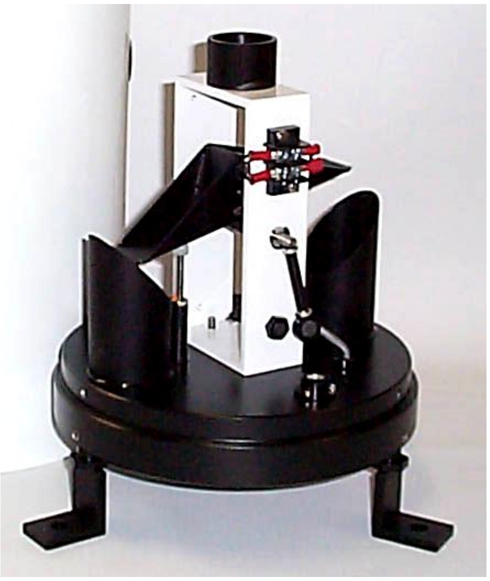


Figure 2.1 Inside view of a state of the art tipping bucket rain gauge (Source: http://www.novalynx.com/images/tipping-bucket-large.jpg).

On 2006, a tipping bucket rain gauge network was installed near the University of Puerto Rico's Mayagüez Campus (UPRM) in western Puerto Rico. It was installed using a Ground Positioning System (GPS) instrument to locate the rain gauges within a Geostationary Operational Environment Satellite (GOES) pixel; an area of 4km x 4km (figure 2.2). The network consists of 28 rain gauges; 16 installed on July 2006, and 12 installed on June 2007 (*Harmsen et al.*, 2008) each one equipped with a data logger that enables rainfall data to be recorded every 5 minutes. The data logger recording frequency can be adjusted (e.g., 10mins, 20mins, 30mins, etc.). The area selected corresponding to a GOES pixel was divided into 16 evenly distributed squares of 1km x 1km (figure 2.3) where the first 16 rain gauges were located using the assistance of a ground positioning system (GPS). The other 12 rain gauges installed on 2007 were distributed within a subwatershed of the Añasco River (*Harmsen et al.,* 2008).

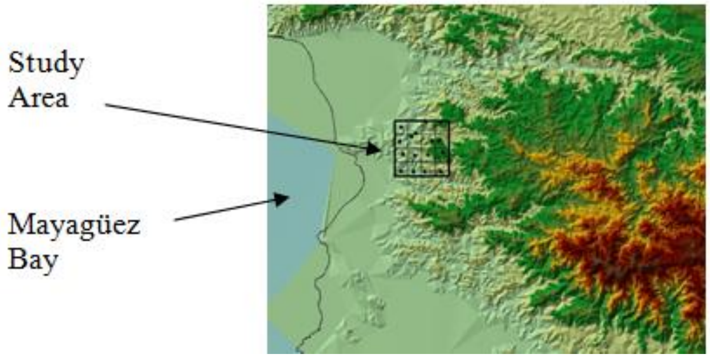


Figure 2.2 Location of rain gauge network installed at western Puerto Rico (Source: *Harmsen et al.*, 2008).

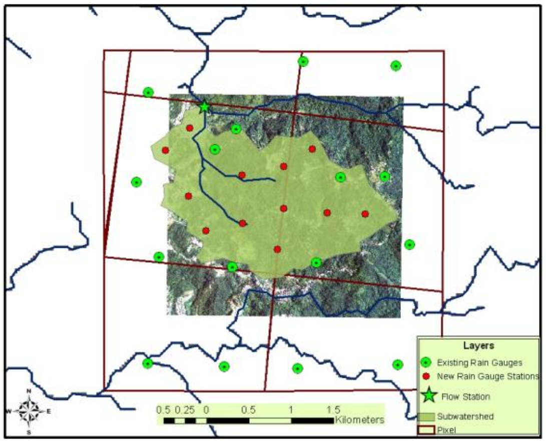


Figure 2.3 Location of rain gauges installed on western Puerto Rico within a GOES pixel (Source: *Harmsen et al.*, 2008).

2.2 GOES

NOAA's Geostationary Operational Environmental Satellites (GOES) are satellites designed to provide remotely sensed information on a near-real time basis for continuous monitoring, but not limited, of the earth atmospheric conditions (*OSO*, 2011). This is possible given that these satellites orbit the earth on a geosynchronous orbit, which enables them to always view the same portion of the earth. GOES satellites are fixed on the geosynchronous plane, which lies at 35,000km (22,300miles) above the earth. This altitude allows the satellites to be able to capture the earth full disk (*OSO*, 2011). NOAA runs two of these geostationary satellites over the equator; GOES- East known as GOES-12 (GOES-13 after April 14, 2010) and GOES-West known as GOES-11. GOES-East is located at the longitude 75W while GOES-West is located on the longitude 135W (*NOASIS*, 2011b).

GOES-12/13 imager (figure 2.4), one of the two instruments aboard of the platform, consists of a five channel (i.e., one visible, four infrared) imaging radiometer designed to sense radiant and solar reflected energy from sampled areas of the earth. The visible channel is centered at 0.65µm; the infrared channels are centered at 3.9µm, 6.9µm, 10.7µm, and 13.3µm. The visible channel has a 1km x 1km resolution while infrared channels have a 4km x 4km resolution, except for the fifth channel (known as 6), which has a resolution of 4km x 8km. This summarized on table 2.5.

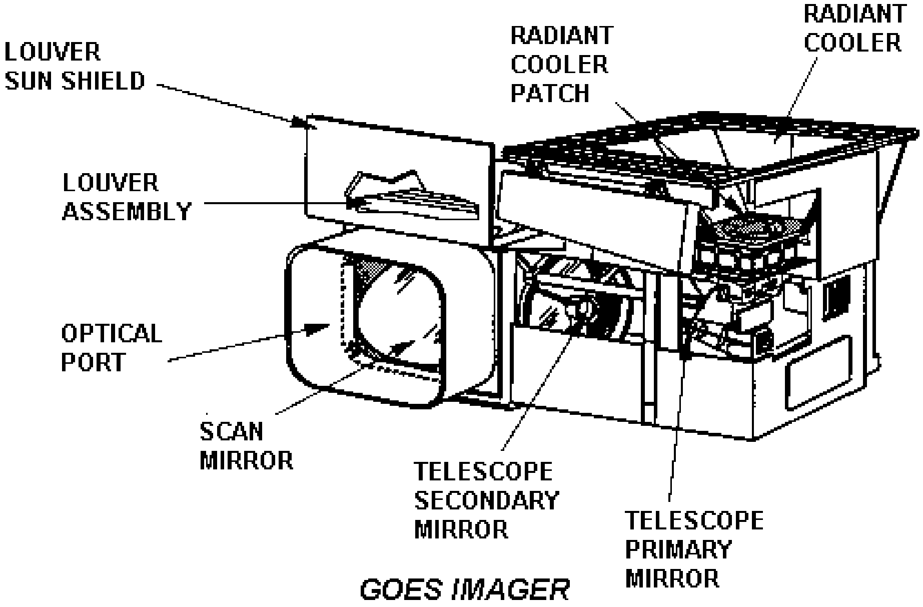


Figure 2.4 GOES Imager (Source: *NOAASIS*, 2011a).

TABLE 2.1 GOES Imager Channels.

Channel	**1**	**2**	**3**	**4**	**6**
Wavelength	0.5 – 0.7µm	3.8 – 4.0µm	5.7 - 7.3µm	10.2 – 11.2µm	12.9 – 13.7µm
Resolution	1km x 1km		4km x 4km		4km x 8km

2.3 NEXRAD

Next Generation Weather Radar system (NEXRAD) consists of 159 Weather Surveillance Radar-1988 Doppler (WSR-88D) sites spread throughout the United States and United States territories (i.e., Guam and Puerto Rico) (figure 2.5). These radars are managed by the National Weather Service (NWS), Air Force Weather Agency (AFWA) and Federal Aviation Administration (FAA) (*NCDC*, 2005). Puerto Rico has one of these NEXRAD radars located at Cayey, Puerto Rico, identified as TJUA (figure 2.6), at the latitude 18.12ºN and longitude 66.08ºW, and at an altitude of 2794 ft (851.6m) (*CWOP* , 2011). It has a frequency of 2.7GHz and a maximum horizontal coverage of 462.5km enabling it to scan the entire island every 6 minutes. Thanks to efforts from the National Server Storms Laboratory (NSSL) from the National Ocean and Atmospheric Administration (NOAA) level II NEXRAD data (i.e., base reflectivity, base velocity, and base spectrum width) for Puerto Rico was collected, distributed and archived (*Kelleher* et al. 2007); level III data (i.e., composite reflectivity, storm mean relative velocity, one-hour precipitation, storm total precipitation, among other processed products) is available since 2000 (*NCDC*, 2005).

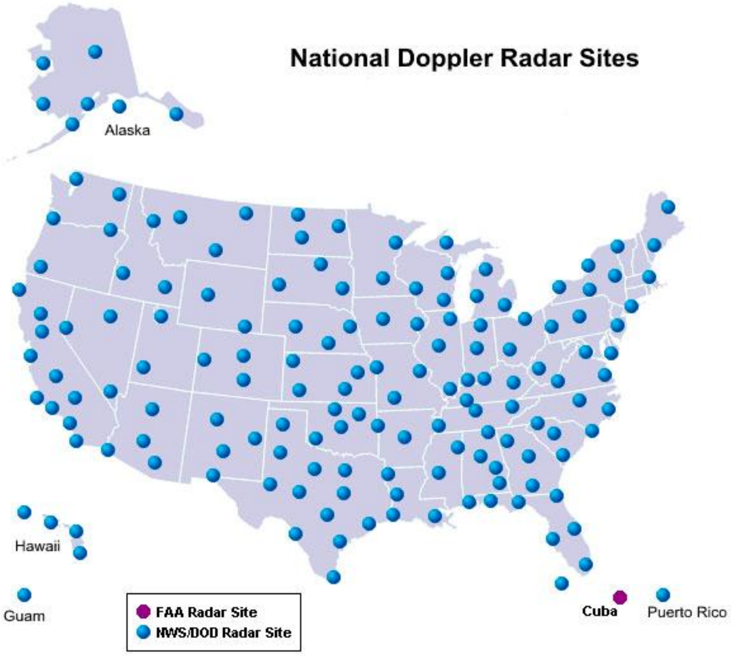


Figure 2.5 NEXRAD sites over U.S. and U.S. Territories (Source: http://noaasis.noaa.gov/NOAASIS/ml/imager.html).

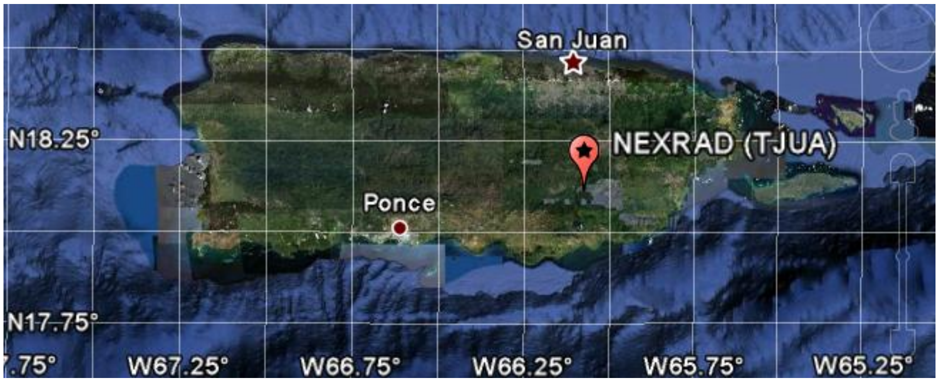


Figure 2.6 Location of NEXRAD (TJUA), Cayey, Puerto Rico.

2.4 Rainfall Estimation Algorithms

Rainfall estimation algorithms can provide rainfall information for those places where other rainfall estimation techniques such as rain gauges and radars cannot, or suffer from poor performance. In the following sections five popular rainfall algorithms will be discussed. Although the discussed algorithms where created for rainfall estimation, cloud rainy pixels detection aspects and classification efforts will be emphasize.

2.4.1 Hydro-Estimator

The National Oceanic and Atmospheric Administration (NOAA) National Environmental Satellite, Data and Information Service (NESDIS) Hydro-Estimator (HE) (*Scofield and Kuligowski*, 2003) is a numerical weather prediction and brightness temperature-based algorithm developed to provide rainfall estimates on a near-real timely matter. It relies on infrared (IR) remotely sensed imagery and National Centers for

Environmental Prediction (NCEP) North American Model (NAM) inputs to detect, estimate, and adjust rain rates. The IR imagery is a product of the GOES-12/13 satellite channel 4 (10.7µm), which provides images every 15 minutes at 4km x 4km resolution. These images can be translated into brightness temperature (T_b) utilized on the HE on a fixed brightness temperature - rain rate relationship (T_b-RR) to estimate rainfall.

Before the HE can derive rain rates it undergoes a rain/no-rain discriminatory process. For this process the HE selects an area of 101 x 101 pixels (400km x 400km) centered on the pixel of interest. The minimum temperature (T_{min}) is obtained from the selected area and used to determine the pixel's surroundings region of interest used for the following steps. If T_{min} is lower than 200K it implies that there is a large convective core and therefore an area of 50 pixels in radius is selected. A T_{min} of 220K or higher implies that there is a smaller convective core and the region selected will consists of a radius of 30 pixels.

Rainfall pixels are defined as those, which T_b is lower than the average of the surroundings. This is achieved utilizing the standardized temperature, given by equation 2.1.

$$Z = \frac{\mu_T - T}{\sigma_T} \qquad \textbf{2.1}$$

Where T is the brightness temperature at the top of the cloud, μ_T is the average brightness temperature of the surroundings of the pixel and σ_T is the standard deviation of the brightness temperatures of the surroundings of the pixel. For negative values of Z it is assumed that the pixel belongs to a cirrus cloud and therefore the rain rate is set to zero. Every pixel with positive value of Z gets assigned a rain rate using a brightness temperature a rain rate relationship obtained through regression (*Scofield and Kuligowski*, 2003).

The rainfall is then adjusted for moisture availability and subcloud evaporation with the Precipitable Water (*PW*) and Relative Humidity (*RH*) NAM files. In areas where the convective equilibrium level temperature is relatively high, rainfall gets enhanced. Finally the combination of winds at 850hPa and digital topography is utilized to enhance and reduce rain rates in up and downslope regions respectively (*Vicente et al.*, 1998).

2.4.2 GMSRA

In 2001, *Ba and Gruber* (2001) presented the GOES Multispectral Rainfall Algorithm (GMSRA). GMSRA is an algorithm that utilizes data from 5 channels of GOES satellite (i.e., visible (0.65µm), near infrared (3.9µm), water vapor (6.9µm), and window channels (10.7 and 12µm)). Channel 1 (0.65 µm) is used when available to select optically thick clouds, channel 2 (3.9µm) is used to get the effective radius of cloud particles during daytime, channel 3 (6.9µm) is used to find clouds with overshooting tops, the channel 4 (10.7µm) is used to get T_b at the top of the cloud, and channel 4 is used in conjunction with channel 5 (12 µm) to estimate temperature (*Ba and Gruber*, 2001). The algorithm is based on brightness temperature for rainfall estimation similar to the GOES

Precipitation Index (*GPI*) (*Arkin and Meinsner*, 1987), Auto Estimator (*AE*) (*Vicente et al*., 1998) and HE (*Scofield and Kuligowski,* 2003). To screen out non-raining clouds GMSRA utilizes the effective radius of cloud particles, and spatial and temporal temperature gradients.

In order to screen clouds the algorithm starts by distinguishing cirrus clouds from active cold convective clouds. In order to distinguish cirrus clouds an empirical method developed by *Adler and Negri* (1988) was modified to work with a smaller area of pixels (i.e., 25 x 25 pixels area). This method computes a slope (Eq. 2.3) and temperature gradient (Eq. 2.2) for every cloud with cloud-top temperatures lower than 250K.

$$G_t = T_{avg} - T_{\min} \qquad 2.2$$

$$S = 0.568(T_{\min} - 217) \qquad 2.3$$

where T_{min} is the minimum temperature of the 25 x 25 pixels area, T_{avg} is the average of a 6 pixels surrounding the current pixel. A large G_t is related to a convective cloud while the opposite means that there is a weak gradient associated with cirrus clouds within the window. Pixels that have a G_t less than S ($G_t < S$) are classified as cirrus clouds and rejected as non-raining clouds. The contrary ($G_t > S$) means the pixels are retained as rainy pixels. From this process it can be seen that the threshold value of G_{t} decreases as the minimum cloud-top temperature decreases, therefore pixel nearby 217K or colder are classified as raining (*Kuligowski*, 2002). The final step on this part of the screening is to verify the brightness temperature difference of the pixel between the 10.7µm channel and the water vapor channel (6.9µm) ($T_{b10.7} – T_{b6.9}$) for clouds with T_b lower than 220K even if they were rejected as non-raining clouds in the previous steps. If the difference is negative (i.e., $(T_{b10.7} - T_{b6.9})<0$ or $T_{b6.9}>T_{b10.7}$) the pixel is retained since they are associated with overshooting tops (*Tjemkes et al.*, 1997). Finally as result of their study, *Ba and Gruber* (2001) added a brightness temperature threshold of 230K during nighttime to preserve the skill of the algorithm. This means that anything above 230K is neglected and classified as non-raining clouds.

During the daytime the GMSRA takes advantage of the availability of the visible channel and the channel 2 (3.9µm) when the radiances are dominated by the solar reflected part, and the effective radius of the pixels can be computed. Pixels with visible reflectance less than 0.40 [W/(m^2Sr^2µm)] are considered as non-raining clouds and associated with cirrus clouds. For those pixels with optically thick clouds (visible reflectance above 0.40 [W/(m^2Sr^2µm)]) the Albedo is computed and the effective radius is retrieved from the lookup tables created by *Rosenfeld and Gutman* (1994), which were tabulated using an inversion of a radiative transfer model (*Nakajima and King*, 1990). The last step is to compute rain rates for those pixels with an effective radius larger than 15µm.

2.4.3 PERSIANN

Due to the inadequate reliability and accuracy of previous rainfall algorithm *Hsu et al.*, (1996) developed a new approach for rainfall estimation. Precipitation Estimation from Remotely Sensed Information Using Artificial Neural Network algorithm (PERSIANN) is an algorithm developed to produce rainfall estimates every 30 minutes at a spatial resolution of 0.25º x 0.25º (28km x 28km). It uses ground-based data collected by the Automated Meteorological Data Acquisition System (AMeDAS), Infrared (IR) information from GOES satellites as well as information from passive microwave rainfall estimates from low-orbital satellites from National Aeronautics and Space Administration (NASA), National Oceanic and Atmospheric Administration (NOAA) and Defense Meteorological Satellite Program (DMSP) low altitude polar-orbital satellites (TRMM, DMSP F-13, F-14, F-15, NOAA-15, -16, 17) (*Ferraro et al.,* 1995, *Weng et al.,* 2003, *Sorooshian et al.,* 2005). The previous data is used to retrieve the following input variables:

- T_1^b - the infrared brightness temperature of the pixel
- $SURF$ - index denoting if the pixel is located over land, coast or ocean
- T_3^b - mean T_b of the 3 x 3 window centered at the target pixel
- STD_3^b - standard deviation of the $T_b$of the 3 x 3 window centered at the target pixel
- T_5^b - mean T_b of the 5 x 5 window centered at the target pixel
- STD_5^b - standard deviation of the $T_b$of the 5 x 5 window centered at the target pixel

These features are fed into Artificial Neural Network (ANN), which input layer consists of a clustering procedure (i.e., Kohonen Self Organizing Feature Map (SOFM) (*Kohonen*, 1982)) called the input layer. The input layer function is to detect and classify patterns in the input data without reference of the output data (*Hsu et al.,* 1997). The output of the latter act as input for the second part of the procedure, called the output layer. On this layer PERSIANN utilizes a supervised ANN; a modified version of the Grossberg linear layer (*Grossberg,* 1969). The function of the output layer is to compute specific rainfall rates for the input patterns classified by the SOFM (input layer). Output layer parameters are adjusted using data from passive microwave rainfall estimates processed from low-orbital satellites (*Sorooshian et al.,* 2000).

2.4.4 PERSIANN-CCS

Throughout the years improvements have been made to the original version of the PERSIANN algorithm. On 2004, (*Hong et al.,* 2004) presented an improvement to the latter in, which instead of direct pixel-to-pixel fitting of infrared cloud images to the rain rates the method adds image segmentation and objective classification methods to process cloud images into a set of disjointed cloud-patch regions. This method is called PERSIAN Cloud Classification System (CCS). PERSIANN-CCS establishes different brightness temperature and rain rates relationships $(T_b - R)$ by following four steps. PERSIANN-CCS first creates patches of clouds using what the developers called, the Incremental Threshold Temperature algorithm (ITT) (*Hong et al.,* 2004). ITT grows regions of clouds by gradually incrementing threshold temperatures. For the region growth, a seed is identified (T_{min}), next the threshold is incremented and extended to neighboring pixels from seeded areas until the border of other seeded areas or cloud-free areas are reached. Then from these patches, coldness, geometric and texture features are extracted:

- Coldness features
    - T_{min} - minimum T_b of a cloud-patch
    - T_{mean} - mean T_b of a cloud-patch
- Geometric features
    - $AREA$ - cloud-patch area
    - SI - cloud-patch shape index
- Texture features ($T_b$texture)
    - STD - standard deviation of cloud-path T_b
    - $MSTD^5$ - mean value of local standard deviation of cloud T_b
    - STD_{std}^5 - standard deviation of local standard deviations of cloud T_b
    - $TOPG$ - cloud cold core gradient of T_b
    - Gray-image texture

It is important to clarify that here the brightness temperature gradient is computed for the cloud patch cold core using equation 2.4. The cloud cold core is defined as the group of pixels surrounding the coldest pixel (T_{min}) with brightness temperatures equal or below $T_{min}+15K$.

$$TOPG = \sum_{i=1}^{N} \frac{15}{\mu(i)} \Big/ N \qquad 2.4$$

Where N is the number of pixels along the border of the $T_{min}+15K$ and $\mu(i)$ is the distance from the border pixel i to the coldest pixel.

On the third step features extracted are fed into a SOFM ANN responsible of classifying the patches created in the first step into classes as done on the original version of PERSIANN. Finally, distinct brightness temperature-rain rate curve are assigned to each class. These assigned curves are calibrated utilizing large amounts of GOES images and their corresponding rain gauge corrected NEXRAD rainfall. Because of the comprehensive cloud-patch features and the ability to address the variability of rainfall distribution in different cloud clusters PERSIAN-CCS outperforms PERSIANN (*Hong et al.,* 2004).

2.4.5 SCaMPR

Self-Calibrating Multivariate Precipitation Retrieval (SCaMPR) (*Kuligowski*, 2002) is a rainfall retrieval algorithm, which combines infrared (*IR*) and microwave (*MW*) data to produce rainfall estimates for fine timescales and short time periods. It was developed to overcome the weakness of both geostationary and microwave satellites. Geostationary satellites can provide continuous IR images but only information from the top of the cloud is provided, which can have a weak relationship with the rain rates. MW satellites provide information that can be related better with rain rates but since they are mounted on low-orbit platforms, information is available 2 times per day.

For rain/no-rain discrimination SCaMPR utilizes a number of different predictors, which include and combine other rainfall algorithms estimates and techniques. Estimates from the Autoestimator (*AE*) (*Vicente et a.,* 1998) and the GMSRA (*Ba and Gruber*,

2001) are used apart as well as in “and” and “or” combinations to screen pixels. If both or one of the algorithms detect rainfall those pixels are retained as rainy pixels. SCaMPR also utilizes brightness temperatures thresholds at 6.9µm, 10.7µm and 12µm to discriminate between rain/no-rain. Predictors considered by SCaMPR are summarized below:

- *AE* rainfall estimates are larger than zero
- *GMSRA* rainfall estimates are larger than zero
- both *AE* and *GMSRA* rainfall estimates are larger than zero
- at least one of AE or GMSRA rainfall estimates are larger than zero
- T_b from channel 3 (6.9µm) is lower than the designed threshold (x)
- T_b from channel 4 (10.7µm) is lower than the designed threshold (x)
- T_b from channel 5 (12.0µm) is lower than the designed threshold (x)
- T_b difference between channel 3 and 4 ($T_{b10.7}$-$T_{b6.9}$) is lower than the designed threshold (x)
- GMSRA’s slope (S) is lower than the designed threshold (x)
- GMSRA’s temperature gradient (G_t) and slope (S) difference is larger than the designed threshold (x)
- T_b difference between channel 3 and 4 ($T_{b10.7}$-$T_{b6.9}$) is lower than the designed threshold (x_1) and T_b difference between channel 5 and 4 ($T_{b10.7}$-$T_{b12.0}$) is lower than the designed threshold (x_2)

The thresholds (*x*'s) in the predictors are adjusted with target data from the Special Sensor Microwave Imager (*SSM/I*) (*Ferraro,* 1997), and $G_t$and S belong to the temperature gradient and slope utilized on the GMSRA. In order to select the best thresholds the Heidke Skill Score (*HSS*) is measured for every predictor when data from SSM/I is available with exception of the AE and GMSRA estimates. The latter is done when one or more GOES images and SSM/I images overlap (*Kuligowski*, 2002).

For rainfall estimation the best estimate predictor is selected by computing a forward regression and obtaining the optimal regression coefficients (slope and intercept). In this step 16 predictors are use; 8 predictors including the AE, GMSRA, $T_{b6.9}$, $T_{b10.7}$, $T_{b12.0}$, G_t-S, S, $T_{b10.7}$-$T_{b6.9}$, and nonlinear transformations for each predictor since some of them are related in a nonlinear matter with rain rates.

2.5 Hydro Estimator Validations

The HE (*Scofield and Kuligowski,* 2003), has undergone a number of validations from 2002 through 2008. In 2002 Daniel Vila and Ines Velasco (*Vila and Velasco,* 2002) performed a validation over Del Plata River basin in Buenos Aires province, Argentina, where they studied a severe storm registered during 23-24 of September of 2001. In their study they found that the HE tends to underestimate rainfall from clouds with cloud top temperatures higher than 215K. Similar studies where performed in PR by Beatriz Cruz (*Cruz-González,* 2006), where she found that the HE performs better when rain rates are accumulated over longer periods of time than it does for instantaneous readings. She also found that the HE tends to overestimates rain rates over PR. In terms of rainfall detection the HE obtained 68%, 48% and 60% in probability of detection (POD), false alarm rate (FAR) and hit rate (HIT). However, another validation performed over PR (*Ramírez-Beltrán et al.,* 2008a), reveals that the HE shows underestimation over a single storm. In the latter the HE exhibited an acceptable Hit Rate (76%) but a low probability of detection (36%). These validations were performed at island-scales and at pixel-scale. In the following sub-section a pixel-scale validation is presented.

2.5.1 HE Pixel-Scale Validation over Western Puerto Rico

Throughout a one year period (i.e., August 2006 – 2007) 62 storms events were registered by a rain gauge network installed in western Puerto Rico located within a GOES pixel (*Harmsen et al.,* 2008). The recorded data was utilized to perform a pixel-scale validation of the HE (*Scofield and Kuligowski*, 2003). For every event, the HE was run to get the rain rate associated with the pixel were the rain gauges were distributed. The rainfall measured by the rain gauge network was compared with the value retrieved from the HE and also with NEXRAD data. Comparisons were made in terms of detection and estimation.

Results obtained from this study suggest that the HE has a low Probability of Detection (*POD*) over western PR (50%). Out of the 62 storms that occurred throughout the year the HE detected half of the events. This may be attributed to the fact that the HE is limited with a 235K brightness temperature threshold. Pixel values obtained from missed rainfall events show that the HE missed those storms were the brightness temperature was higher than the algorithm's threshold (figure 2.7 and 2.8). Figure 2.9 shows the rainfall estimation measured by the rain gauge network installed in western PR as well as those obtained from NEXRAD.

It was also observed in this study that large rainfall spatial variability can occur within a single GOES pixel (i.e., 4 x 4km area). It is argued that comparing or validating Quantitative Precipitations Estimation (*QPE*) methods using only one rain gauge for each pixel from existing networks like those from the U.S. Geological Survey may be inadequate. Given the high spatial variability observed in Puerto Rico, the measurement of rainfall obtained will depend on the location of the rain gauge within the GOES pixel (*Harmsen et al.*, 2008).

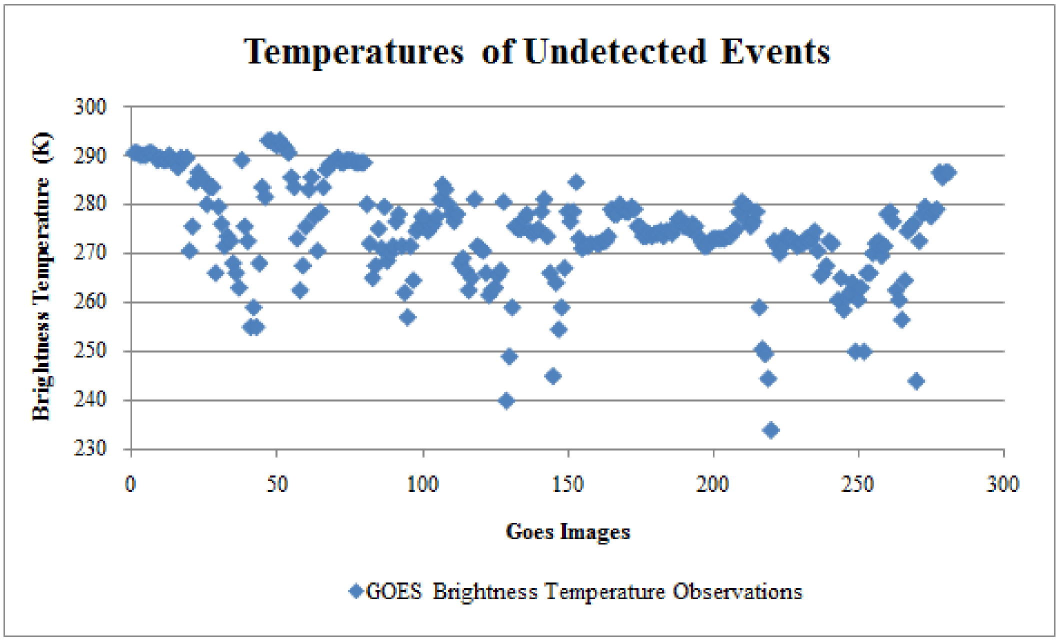


Figure 2.7Cloud top temperature values associated with storms in, which the HE did not detect rainfall.

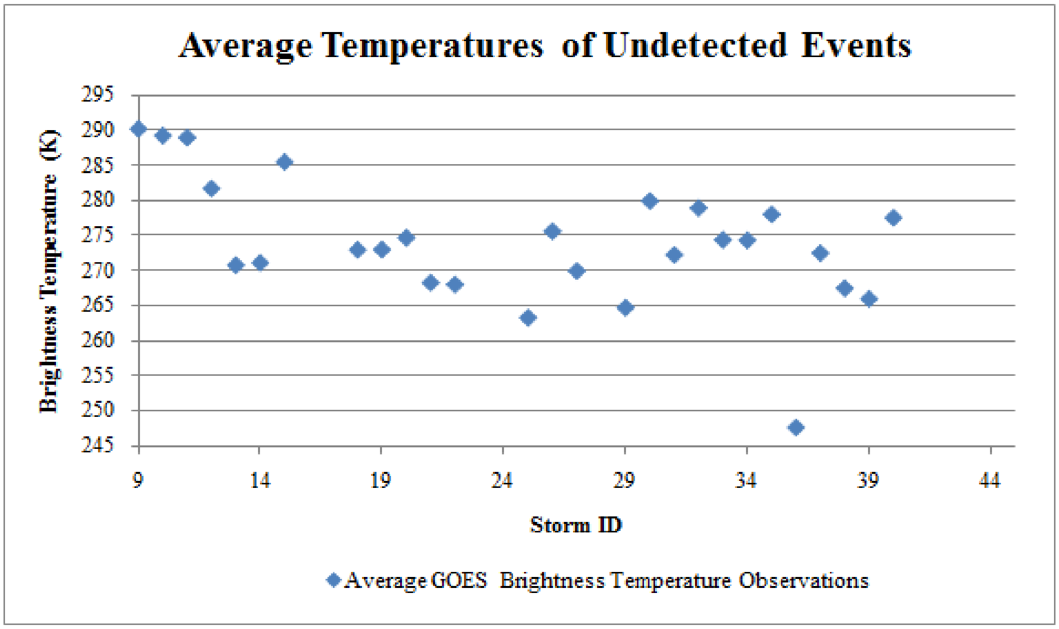


Figure 2.8 Average cloud top temperature values associated with storms in, which the HE did not detect rainfall.

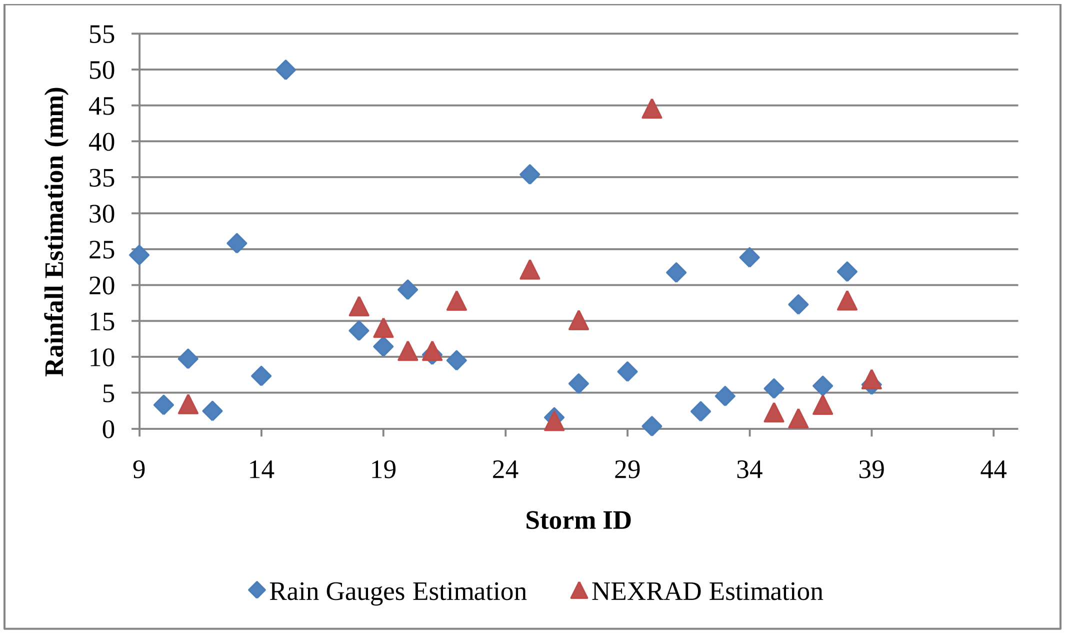


Figure 2.9 NEXRAD and Rain Gauges rainfall estimation associated with storms in, which the HE did not detect rainfall.

3 Methodology

Brightness temperature-based algorithms fail to detect rainfall produced by clouds with warm tops. Some of them use a temperature threshold to screen out non-raining clouds, other algorithms only use one remotely sensed product, but most importantly some of them only use one T_b rain rate relationship. Over PR this causes algorithms like the HE to have a low probability of detection. This chapter presents the methodology used in this study to improve rainfall detection over PR.

3.1 Cloud Classification System

To improve rainfall detection over PR a new rainfall detection algorithm was developed. The new algorithm consists of performing supervised classification utilizing the maximum likelihood method (*MAL*), an algorithm that has been used on other remote sensing applications such as: multispectral lands cover classification (*Schott*, 2007). The classification is done utilizing 4 classes previously defined. For each class more than one remotely sensed feature were obtained from the GOES satellite. The following sections present the methodology behind the cloud classification system. The cloud classification system algorithm's MATLAB code is presented in appendix A. A high level flowchart of the algorithm is presented on appendix B.

3.1.1 Maximum Likelihood

Relying only on T_b to improve rainfall detection over PR is insufficient for accurate rainfall detection. This is the reason why algorithms like the GMSRA and SCAMPR take into account other information and not just T_b. It is possible to obtain various features from GOES 12 to perform a classification. If a classification is to be done utilizing more than one channel on different wavelengths a multivariate classification method is needed. *MAL* provides this capability.

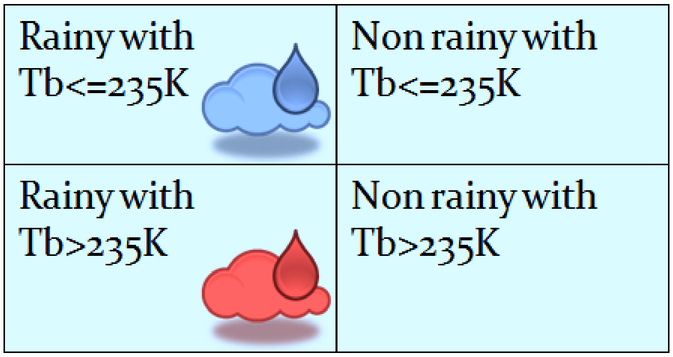
MAL is a multivariate classification method used on multispectral applications such as land cover classification. On this method the pixel is assigned to whatever class it has the highest probability for the spectral vector associated with the pixel (*Schott*, 2007); the vector being formed by all the features extracted from the remote sensing platform.

The MAL assumes that the variables follow a Multivariate Normal Distribution; and therefore, the likelihood function to be maximized is given by Eq. 3.1, where $g_i(x_m)$ is the likelihood of x_m belonging to the class i, x_m is a vector of features to be classified, μ_i is the centroid of the class i, l is the number of features associated with the vector x, S_i is the covariance matrix of the class i *and* W_i represents the list of variables in class i.

$$g_i(x_m) = \frac{1}{(2\pi)^{\frac{l}{2}} |S_i|^{\frac{1}{2}}} e^{\left[-\frac{1}{2}(x_m - \mu_i)^T S_i^{-1}(x_m - \mu_i)\right]}; \; x_m \in W_i \; \mathit{if} \; g_i(x_m) > g_j(x_m) \;\; \mathit{for\ all} \; j \neq i \quad \mathbf{3.1}$$

3.1.2 Rainfall defined classes

The MAL is a supervised classification method. This means it needs to be trained, and therefore the classes in, which it will classify the pixels have to be previously defined. For this, the rainfall detection of National Weather Service's (NWS) NEXRAD radar located on Cayey and GOES features from multiple channels were used. In order to be able to generalize, only 4 classes were defined. First a T_b of 235K was used as threshold to classify clouds as warm and cold clouds to form 2 major classes. Cloud tops with T_b below and/or equal to 235K were identified as cold clouds and those with T_b above 235K were identified as warm clouds. Finally using the rainfall detection provided by NEXRAD the previously defined classes were divided 4 new classes (figure 3.1): (1) Rainy clouds with $T_b \leq 235\text{K}$, (2) Rainy clouds with $T_b > 235\text{K}$, (3) Non rainy clouds with $T_b \leq 235\text{K}$, and (4) Non rainy clouds with $T_b > 235\text{K}$. For each class a centroid was computed and then used to train the classifier. Centroids are vectors, which consist of the means of every feature in the class.



Rainy with Tb<=235K	Non rainy with Tb<=235K
Rainy with Tb>235K	Non rainy with Tb>235K

Figure 3.1 Classes of clouds defined using NEXRAD.

3.1.3 Cloud Classification System Training Data Set

For the training data selection images from rainfall events that occurred from November 2003 to October 2007 were selected. NEXRAD detection information was used to separate rainy pixels from non-rainy pixels on the images available from the first day of the storm. Once the pixels were identified as rainy or non-rainy, pixels that could cause ambiguity where removed from the training data. In order to ensure the latter, rainy pixels were used on the training data set if it was located in the center of 8 rainy pixels and likewise, a non-rainy pixel was used if it was located in the center of 8 non-rainy pixels. This is shown in figure (3.2).

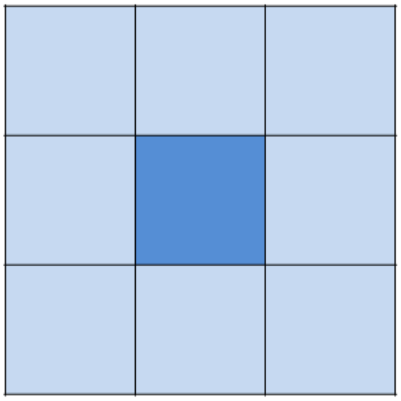


Figure 3.2 Representation of the pixel in question (dark) and the eight surrounding pixels used to determine if it is a rainy or no-rainy pixel.

After ambiguous pixels were removed from the training data set, GOES-12 brightness temperature information from channel 4 (10.7µm) was used to create data sets for each of the 4 classes previously identified. Once the data set were created, centroids were computed to represent each class. They were then used for the supervised classification.

3.2 Feature Selection Process

An array of variables can be derived from GOES data. Some of these variables are: (1) Visible Reflectance centered at 0.65µm (only available at daytime), (2) Albedo centered at 3.9µm (only available at daytime), (3) Brightness Temperature (T_b) centered at 3.9µm, 6.9 µm, and 10.7, and (4) Brightness Temperature Differences (*BTD*) between channels (e.g., $T_{b3.9\mu m}$-$T_{b10.7\mu m,}$ *and* $T_{b6.9\mu m}$-$T_{b10.7\mu m}$). These are shown in Figure 3.1. In order to identify, which remotely sensed variable(s) may help to improve detection over PR, a classification system was designed and assessed. For this the classifier was run for 5 storms, utilizing several feature vectors, each vector with a different combination of features (i.e., [$T_{b3.9\mu m}$-$T_{b10.7\mu m,}$ $T_{b6.9\mu m}$-$T_{b10.7\mu m}$], [$T_{b3.9\mu m}$-$T_{b10.7\mu m,}$ *Albedo* $T_{b3.9\mu m}$], [$T_{b3.9\mu m}$-$T_{b10.7\mu m,}$ $T_{b10.7\mu m}$], etc.).

Features used during nighttime, when visible reflectance and albedo are not present, are the brightness temperature difference between channels 2 (3.9µm) and channel 4 (6.9µm), the brightness temperature difference between channels 3 (6.9µm) and channel 4 (6.9µm), and brightness temperatures (T_b) centered at 3.9µm, 6.9 µm, and 10.7.

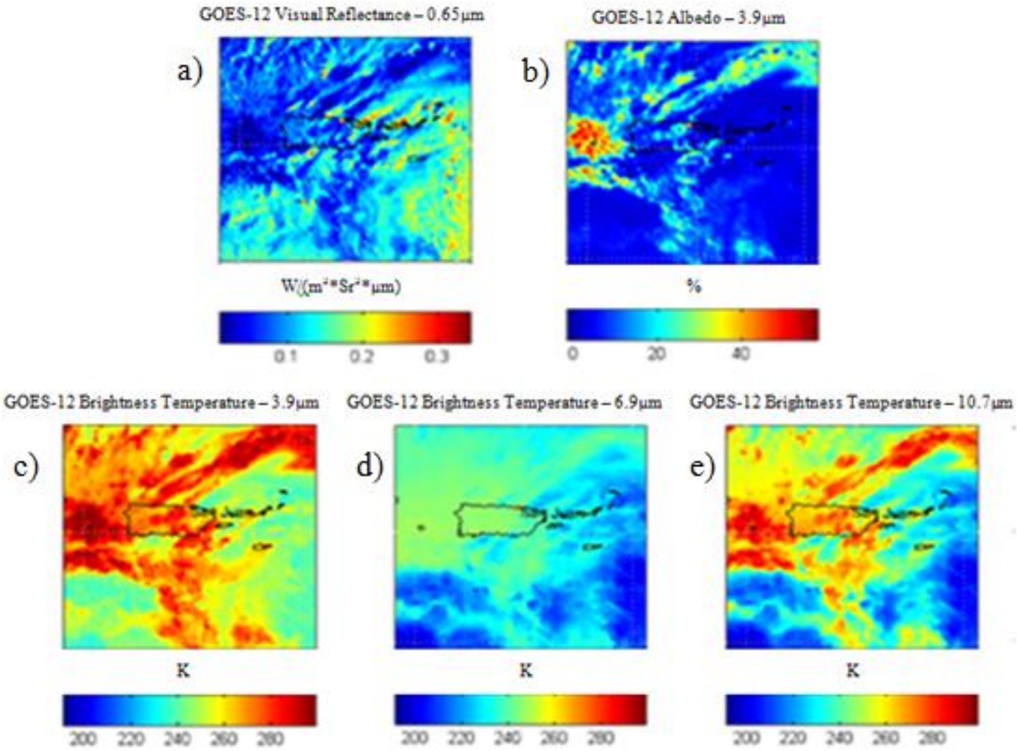

GOES-12 Visual Reflectance – 0.65µm
a)
W/(m^-2*Sr^-1*µm)
0.1
0.2
0.3
GOES-12 Albedo – 3.9µm
b)
%
0
20
40
GOES-12 Brightness Temperature – 3.9µm
c)
K
200
220
240
260
280
GOES-12 Brightness Temperature – 6.9µm
d)
K
200
220
240
260
280
GOES-12 Brightness Temperature – 10.7µm
e)
K
200
220
240
260
280

Figure 3.3 Features available for extraction from GOES-12 for the vicinity of Puerto Rico: a) Visible reflectance (0.65µm), b) Albedo (3.9µm), and c) Brightness Temperatures from channel 2 (3.9µm), c) channel 3 (6.9µm) and d) channel 4 (10.7µm).

3.2.1 Remotely Sensed Products

In order to properly determine the rainfall produced by a cloud, it is imperative that those pixels from the satellite image capable of producing rain are detected. Brightness temperature alone might not be the best way of detecting rainfall since warm clouds (i.e., T_b > 235K) are able to produce rainfall over PR. Having this in mind, MATLAB scripts were developed to convert GOES-12 GVAR-Counts to several features.

3.2.1.1 Visible Reflectance

Reflectance can be defined as the ratio of reflected visible radiation to the incident visible radiation (*Schott*, 2007). The GOES Imager experiences continuous degradation while in orbit, and therefore, the pre-launch calibration needs to be corrected with a post-launch calibration (*Wu and Sun*, 2005). Visible reflectance from GOES (figure 3.4) is computed as:

$$V_{ref} = a * \exp(bt) * A_{pre} \qquad \mathbf{3.2}$$

where V_{ref} is the visible reflectance by post-launched calibration, $(a * \exp(bt))$ is the post-launched calibration term, t is the time in years from the date when the satellite was launched to the date of the image, A_{pre} is the visible reflectance by pre-launched calibration, and a and b are calibration coefficients. Visible reflectance by pre-launched calibration is defined as:

$$A_{pre} = k(X - X_{space}) \qquad \mathbf{3.3}$$

where k is a calibration coefficient, X is the 10-bit count, and X_{space} is the instrument response to space scene where signal is expected to be zero (should always be 29). The MATLAB code used to derive visible reflectance from GOES images is presented in appendix C.

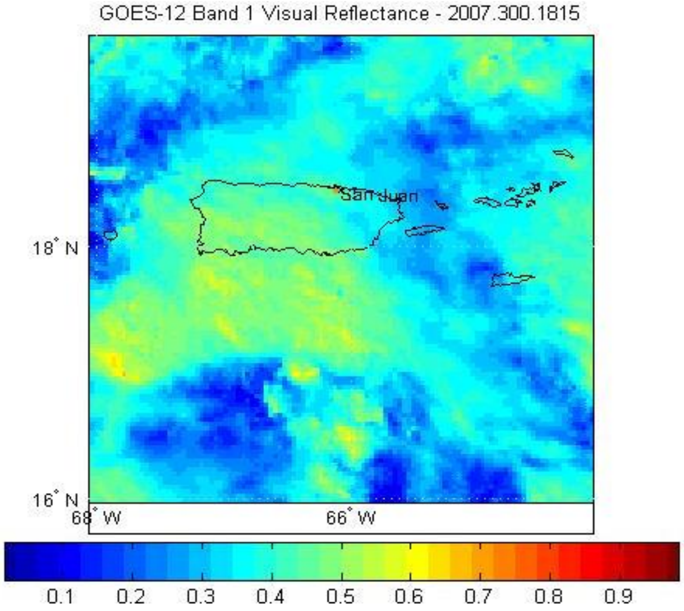


Figure 3.4 Visible Reflectance image from October 27, 2008 (18:15 UTC)

3.2.1.2 Brightness Temperature

Brightness temperature can be defined as a descriptive measure of radiation in terms of the temperature of a hypothetical blackbody emitting an identical amount of radiation at the same wavelength (*AMS*, 2011). Brightness temperature is computed for GOES-12/13 channels 2 (figure 5), 3 (figure 6), and 4 (figure 7) (3.9, 6.9 and 10.7 µm) as follows:

$$\begin{aligned} T_b &= 330 - 0.5 * count \qquad &; count < 176 \\ T_b &= 418 - count \qquad &; count \geq 176 \end{aligned} \tag{3.4}$$

where *count* is the digital number registered by the satellites sensor. The MATLAB code used to derive brightness temperature from GOES images is presented in appendix D.

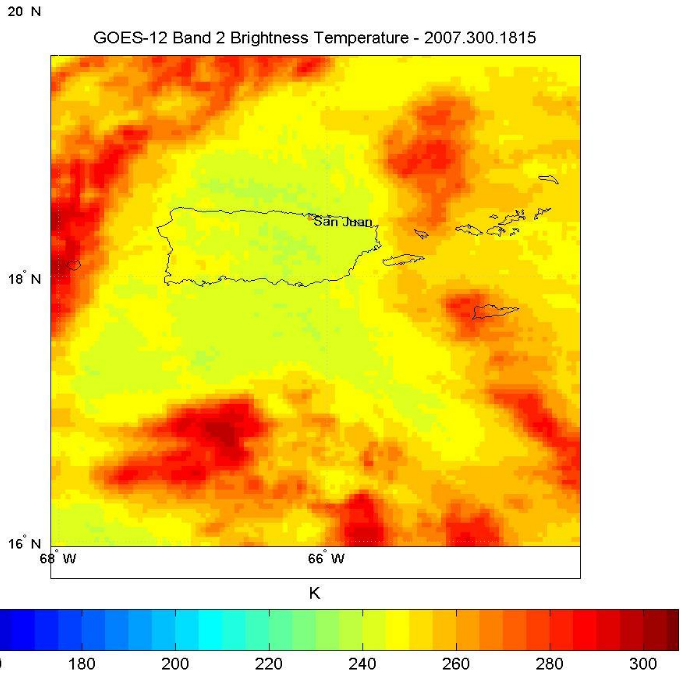


Figure 3.5 Channel 2 Brightness Temperature (K) from October 27, 2008 (18:15 UTC)

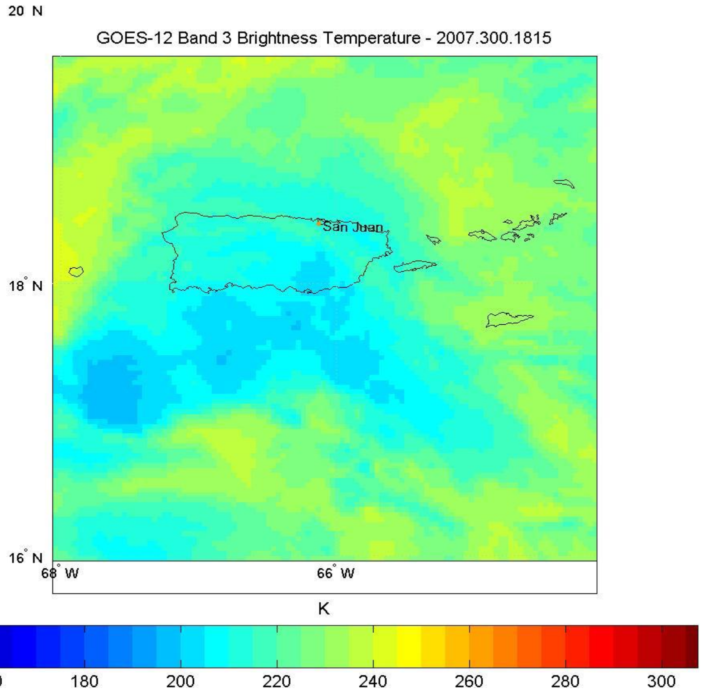


Figure 3.6 Channel 3 Brightness Temperature (K) from October 27, 2008 (18:15 UTC)

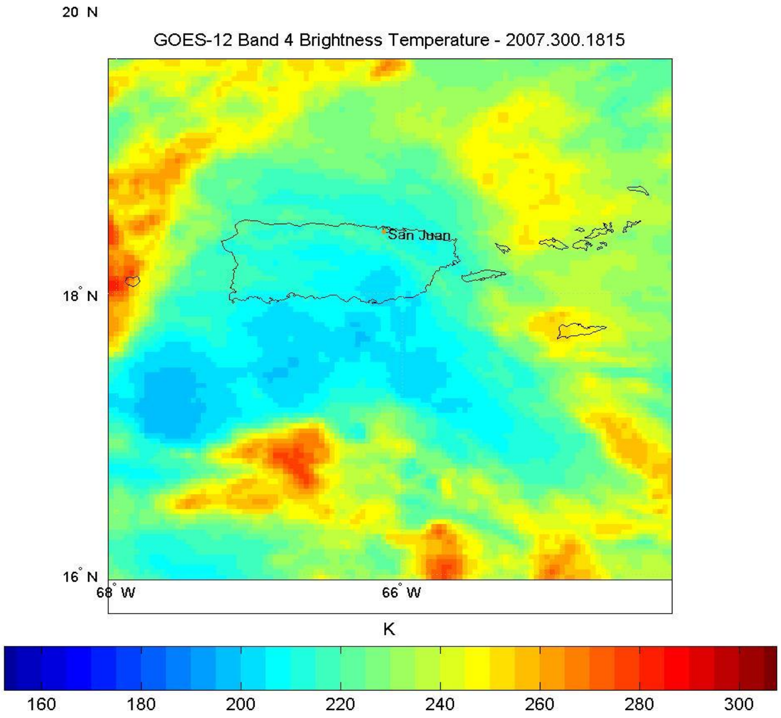


Figure 3.7 Channel 4 Brightness Temperature (K) from October 27, 2008 (18:15 UTC)

3.2.1.3 Albedo

In 2008, Lindsey and Grasso (*Lindsey and Grasso*, 2008), presented an algorithm to extract the effective radius of clouds centered at 3.9µm, utilizing data from GOES-12, channels 2 and 4 and based on albedo plus the scattering and solar zenith angle. Unfortunately, the lookup tables do not work for PR conditions. The algorithm was developed to work with clouds with brightness temperatures lower than 233K. Therefore, for locations where warm cloud rainfall occurs, like PR, the effective radius estimation gets saturated (i.e., most of the pixels receive the highest value on the tables).

Due to the latter situation, the albedo portion was extracted to and used in the classification algorithm to provide an indirect relationship for estimating effective radius.

The algorithm starts by converting GVAR-Counts from each channel into brightness temperatures. Once the brightness temperatures are computed, then the algorithm calculates geometric parameters needed to perform the correct calculations based on date and time. The parameters computed are: a) solar zenith angle, b) scattering angle, c) GOES platform zenith angle, and d) relative azimuth angle (difference between the solar and satellite azimuth angles) (figure 3.8).

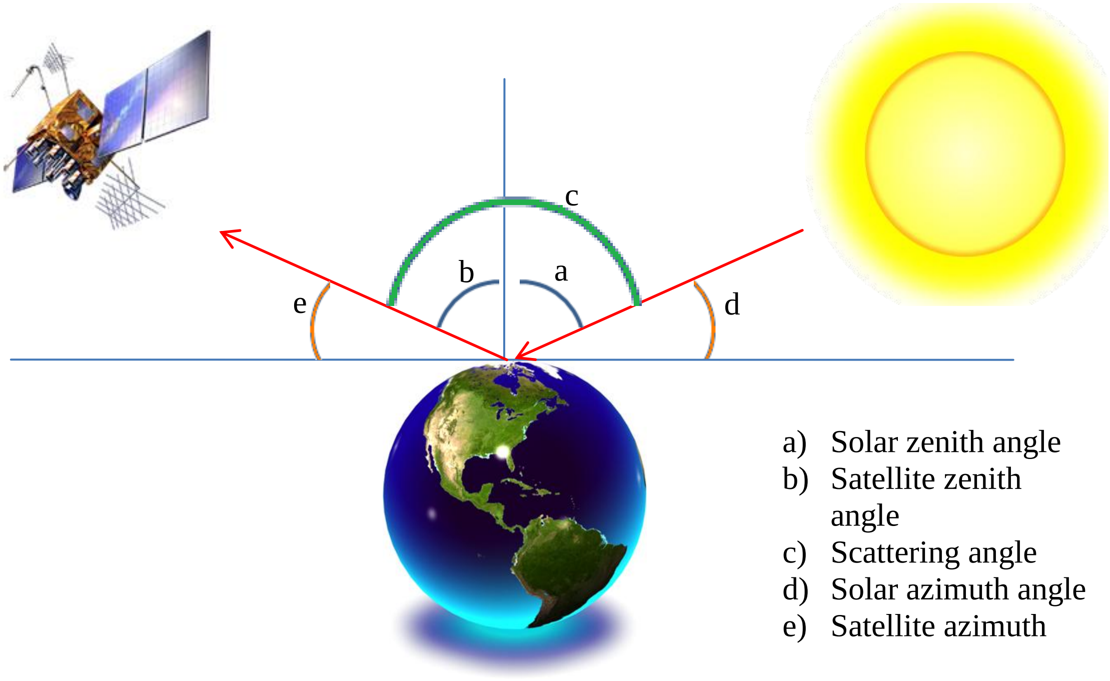


Figure 3.8 Geometric parameters.

Once the geometric parameters are computed for every pixel in the image, the albedo gets computed as follows:

$$a = \frac{R_{3.9} - R_{e3.9}}{S - R_{e3.9}} \qquad \mathbf{3.5}$$

where a is the albedo at 3.9 microns, $R_{3.9}$ is the observed radiance from channel 2, S is the solar irradiance of GOES 12 channel 2, and $R_{e3.9}$ is the equivalent black body emitted thermal radiation at 3.9 microns for cloud to temperature T_c. $R_{3.9}$ or total radiance is defined as:

$$R_{3.9} = Planck(waveNumber_{3.9}, T_{b3.9}) \qquad \mathbf{3.6}$$

where $R_{3.9}$ is the total radiance at the channel centered at 3.9µm, waveNumber$_{3.9}$ is the wave length number in cm^{-1} and, $T_{b3.9}$ is the brightness temperature at 3.9µm. $R_{e3.9}$ or estimated blackbody radiance is defined as:

$$R_{3.9} = Planck(waveNumber_{3.9}, T_{b10.7}) \qquad \mathbf{3.7}$$

where $R_{e3.9}$ is the estimated blackbody radiance at 3.9µm, waveNumber$_{3.9}$ is the wave length number in cm^{-1} and, $T_{b3.9}$ is the brightness temperature at 3.9µm. *Planck(waveNumber,T_b)* in (Eq. 3.6) and (Eq. 3.8) refers to the Planck's equation. The same is used to calculate radiances. Planck's equation is defined as:

$$I(\lambda, T) = \frac{2hv^2}{\lambda^5} \frac{1}{\exp\left(\frac{hv}{\lambda kT}\right) - 1} \qquad \mathbf{3.8}$$

where $I(waveNumber_{3.9}, T_{b3.9}) = R_{3.9}$, and $I(waveNumber_{3.9}, T_{b10.7}) = R_{e3.9}$, and $T = T_b$., v

v is the speed of light (299,792,458m/s), h is the Plank constant {6.62606896(33)x10^{-34} , J.s}, and k is the Boltzmann constant {1.3806504(24)x10^{-23} J.K^{-1}}. The MATLAB code used to derive albedo from GOES images is presented in appendix E.

3.3 Data Set

Five rain events were selected to measure the performance of the new rainfall detection algorithm over PR (table 3.1). The selected events were heavy storms that occurred from 2003 to 2007, which created damage to infrastructure and/or humans. Information was retrieved from GOES-12, NEXRAD and the HE. From GOES, images where obtained from channels 1, 2, 3 and 4 to get Visible Reflectance 0.65µm, Brightness Temperatures at 3.9, 6.9 an 10.7µm, and albedo centered at 3.9µm. Rain rates were obtained from NEXRAD and the HE and converted into rainfall detection. Pixel with no rain where given a binary value of 0 and 1 for pixels with rain.

TABLE 3.1 Rainfall Events.

Event Date	**Description**
November 11-12, 2003	driven by cyclone, which stalls and breaks down into a trough
December 5-8, 2003	driven by the merging of a tropical vortex with a mid latitude cold front
April 19-24, 2005	driven by a sub-tropical trough that is poorly organized
May 17, 2005	driven by a south west trough coming from Caribbean sea
October 27-30, 2007	driven by an easterly wave passing by PR during this time

3.4 Performance Evaluation

The classifier's discrete performance (detection) is to be measured for every step mentioned in previous section based on hit rate (*HIT*), bias, probability of detection (*POD*), and false alarm rate (*FAR*). Once the discrete performance is computed for all the runs, a performance index (Eq. 3.13) defined by *Ramirez-Beltran et al*. (2009) will be utilized to obtain an overall performance. The index takes into account the FAR, POD and HIT. An index of 0 indicate a perfect performance (FAR=0, POD=1, HIT=1) while 1 states the contrary or the worst case scenario (FAR=1, POD=0, HIT=0).

$$POD = \frac{a}{a+c} \qquad \textbf{3.9}$$

$$FAR = \frac{b}{a+b} \qquad \textbf{3.10}$$

$$HIT = \frac{a+d}{a+b+c+d} \qquad \textbf{3.11}$$

$$BIAS = \frac{a+b}{a+c} \qquad \textbf{3.12}$$

$$index = \frac{FAR - POD - HIT + 2}{3} \qquad \textbf{3.13}$$

The terms *a* through *d* utilized for each equation are extracted from a contingency table (table 3.2). The table takes into account the detection of the classifier and assesses it with the detection of NEXRAD, which is used as ground truth. A similar table (table

3.3) was created to measure the performance of the HE to be able to contrast both algorithms performance in terms of detection.

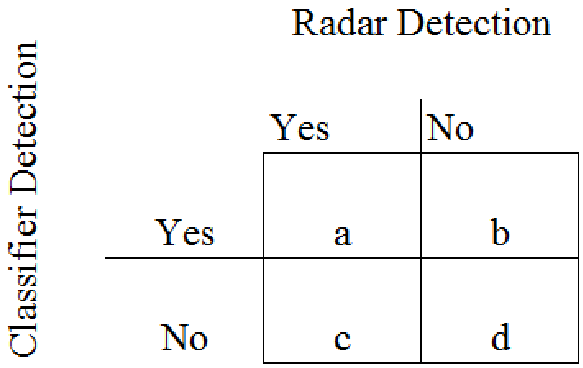
To assess how well the cloud classification system detects rainfall produced by warm clouds, the warm cloud rainfall detection percentage will be obtained. The latter is defined as the ratio of pixels that produced rainfall and have a brightness temperature above 235K detected by the cloud classification system to those detected by the NEXRAD, expressed as a percentage. Equation 3.14 shows how this metric is obtained.

$$\%WarmCloudRainDetecti\,on = \frac{n\_ByA\lg}{n\_ByNEXRAD}*100 \qquad \mathbf{3.14}$$

Where *n_ByAlg* is the number of pixels that produced rainfall and have a brightness temperature above 235K and were detected by the cloud classification system and *n_ByNEXRAD* is the number of pixels that produced rainfall and have brightness temperature above 235K detected by NEXRAD. The same metric is used for the HE to determine if using a cloud classification system with multiple remotely sensed variables improves warm cloud rainfall detection.

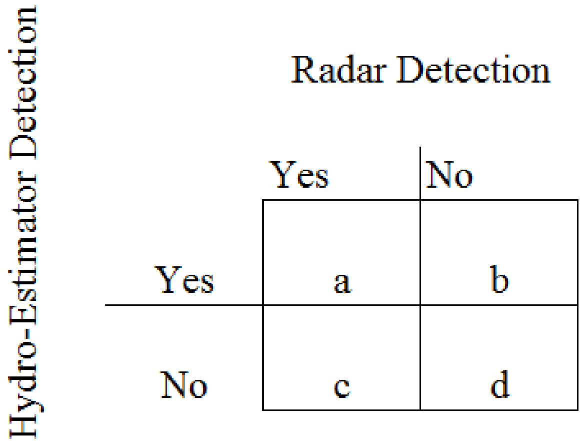
Once the cloud classification system was run for all the storms, a graphical comparison was made to visually compare the rainfall detection performance of both the HE and the cloud classification system with NEXRAD.

TABLE 3.2 Classifier detection contingency table.



		Radar Detection	
		Yes	No
Classifier Detection	Yes	a	b
	No	c	d

TABLE 3.3 Hydro-Estimator detection contingency table.



		Radar Detection	
		Yes	No
Hydro-Estimator Detection	Yes	a	b
	No	c	d

4 Results

Five heavy storms that occurred during 2003 to 2007 were selected for a discrete comparison between the Hydro-Estimator (HE) and the developed cloud classification system. This chapter presents the results obtained on the feature selection process and the discrete comparison between the cloud classification and the HE during daytime and nighttime.

4.1 Selected Features

In order to identify the features that best describe rainfall occurrence over PR, the five selected storms were divided in half. Half of the images available for each storm were used for calibration, and the other half was used for validation. To derive centroids for each of the four classes, NEXRAD rainfall detection information was used. Five set of centroids were obtained; one from each storm (5 storms x 5 centroid sets). The centroids obtained were used to perform validation with the observations with the second half of every storm (i.e., centroids selected for November 2003 were used to perform validation with every storm and equally with every centroid obtained). The validation consisted of running the cloud classification system using the centroids obtained from the first half of every storm and testing every possible feature combination (26 possible combinations). The latter resulted in 650 runs, which were used to determine the best feature combination as well as the best centroids set. It is important to emphasize that this step is only done once and only to determine the classification system parameters at the development phase.

During daytime the remotely sensed features selected with the best overall performance were the combination of visible reflectance centered at 0.65µm, brightness temperature difference between channels 2 (3.9µm) and 4 (10.7µm), brightness temperature from channel 4 (10.7µm) and Albedo centered at 3.9µm. Table 4.1 presents the five feature combinations, out 26 possible combinations, that obtained the best performance in terms of detection.

TABLE 4.1 Average performance of the best features for rainfall detection during daytime.

Feature Combination	$\overline{INDEX}$
$VisRef_{0.65\mu m}$, $T_{b3.9\mu m}$-$T_{b10.7\mu m}$, $T_{b10.7\mu m}$, $Alb_{3.9\mu m}$	0.433
$VisRef_{0.65\mu m}$, $T_{b6.9\mu m}$-$T_{b10.7\mu m}$	0.446
$VisRef_{0.65\mu m}$, $Alb_{3.9\mu m}$	0.448
$VisRef_{0.65\mu m}$, $T_{b3.9\mu m}$-$T_{b10.7\mu m}$, $T_{b6.9\mu m}$-$T_{b10.7\mu m}$, $T_{b10.7\mu m}$, $Alb_{3.9\mu m}$	0.453
$VisRef_{0.65\mu m}$, $T_{b6.9\mu m}$-$T_{b10.7\mu m}$, $Alb_{3.9\mu m}$	0.464

During nighttime when visible reflectance and albedo are not available, the remotely sensed features selected with the best overall performance were the combination of the brightness temperature difference between channels 2 (3.9µm) and 4 (10.7µm) and the brightness temperature difference between channels 3 (6.9µm) and 4 (10.7µm). Table 4.2 shows the five feature combinations, which obtained the best performance in terms of detection.

TABLE 4.2 Average performance of the best features for rainfall detection during nighttime.

Feature Combination	$\overline{INDEX}$
$T_{b3.9\mu m}$-$T_{b10.7\mu m}$, $T_{b6.9\mu m}$-$T_{b10.7\mu m}$	0.356
$T_{b3.9\mu m}$, $T_{b6.9\mu m}$-$T_{b10.7\mu m}$	0.368
$T_{b3.9\mu m}$, $T_{b6.9\mu m}$, $T_{b3.9\mu m}$-$T_{b10.7\mu m}$, $T_{b6.9\mu m}$-$T_{b10.7\mu m}$	0.376
$T_{b10.7\mu m}$, $T_{b3.9\mu m}$-$T_{b10.7\mu m}$	0.387
$T_{b3.9\mu m}$, $T_{b3.9\mu m}$-$T_{b10.7\mu m}$	0.387

4.2 Selected Centroid Set

During daytime the centroid set that produced the best performance results were those obtained from the storm of October 2007. This centroid set obtained an average performance index of 0.35. Rainy clouds with brightness temperature below 235K are defined as those that average visible reflectance centered at 0.65µm of 0.56 [W/(m^{2}Sr^{2}µm)], a difference of 30.39K between brightness temperature from channel 2 (3.9µm) and channel 4 (10.7µm), a brightness temperature of 215.08K on channel 4 (10.7µm), and an albedo of 1.79%. Rainy clouds with brightness temperature higher than 235K are defined as those that average visible reflectance centered at 0.65µm of 0.61 [W/(m^{2}Sr^{2}µm)], a difference of 16.03K between brightness temperature from channel 2 (3.9µm) and channel 4 (10.7µm), a brightness temperature of 241.49K on channel 4 (10.7µm), and an albedo of 2.22%. Similar to the feature selection this step is only done once, only to determine the classification system parameters at the development phase.

Non-rainy clouds with brightness temperature below 235K are defined as those that average visible reflectance centered at 0.65µm of 0.48 [W/(m^{2}Sr^{2}µm)], a difference of 26.30K between brightness temperature from channel 2 (3.9µm) and channel 4 (10.7µm), a brightness temperature of 223.09 K on channel 4 (10.7µm), and an albedo of 2.15%. Non-rainy clouds with brightness temperature higher than 235K are defined as those that average visible reflectance centered at 0.65µm of 0.36 [W/(m^{2}Sr^{2}µm)], a difference of 19.33K between brightness temperature from channel 2 (3.9µm) and channel 4 (10.7µm), a brightness temperature of 251.93K on channel 4 (10.7µm), and an albedo of 5.88%. Table 4.3 summarizes the centroid values for every class.

TABLE 4.3 Centroids selected for daytime rainfall detection.

Class Description	$\boldsymbol{VisRef_{0.65\mu m}}$	$\boldsymbol{T_{b3.9\mu m}\text{-}T_{b10.7\mu m}}$	$\boldsymbol{T_{b10.7\mu m}}$	$\boldsymbol{Alb_{3.9\mu m}}$
Rainy clouds T_b<235	0.56	30.39	215.08	1.79
Rainy clouds T_b>235	0.61	16.03	241.49	2.22
Non-rainy clouds T_b<235	0.48	26.30	223.09	2.15
Non-rainy clouds T_b>235	0.36	19.33	251.93	5.88

During nighttime the centroids with the best detection performance were those obtained from the storm of May 2005 with an average index of 0.31. Rainy clouds with brightness temperature below 235K are defined as those that average a difference of 1.04 K between brightness temperature from channel 2 (3.9µm) and channel 4 (10.7µm), and an -0.09 K difference between brightness temperature from channel 3 (6.9µm) and channel 4 (10.7µm). Rainy clouds with brightness temperature higher than 235K are defined as those that average a difference of 3.82K between brightness temperature from channel 2 (3.9µm) and channel 4 (10.7µm), and an –7.19K difference between brightness temperature from channel 3 (6.9µm) and channel 4 (10.7µm).

Non-rainy clouds with brightness temperature below 235K are defined as those that average a difference of 5.77K between brightness temperature from channel 2 (3.9µm) and channel 4 (10.7µm), and an -1.62K difference between brightness temperature from channel 3 (6.9µm) and channel 4 (10.7µm). Non-rainy clouds with brightness temperature higher than 235K are defined as those that average a difference of 10.18K between brightness temperature from channel 2 (3.9µm) and channel 4 (10.7µm), and an –16.28K difference between brightness temperature from channel 3 (6.9µm) and channel 4 (10.7µm). Table 4.4 summarizes the centroids selected for rainfall detection during nighttime.

TABLE 4.4 Centroids selected for nighttime rainfall detection.

Class Description	$T_{b3.9\mu m}$-$T_{b10.7\mu m}$	$T_{b6.9\mu m}$-$T_{b10.7\mu m}$
Rainy clouds T_b<235	1.04	-0.09
Rainy clouds T_b>235	3.82	-7.19
Non-rainy clouds T_b<235	5.77	-1.62
Non-rainy clouds T_b>235	10.18	-16.28

4.3 Discrete Performance Comparison

Once the centroids and the features that best describe rainfall occurrence over Puerto Rico were selected, both the cloud classification algorithm and the HE were run for all of the storms. The cloud classification system was run using the best configuration obtained. The discrete performance of both algorithms was obtained to compare them with one another. The validation of performance was measured during daytime and nighttime using information from NEXRAD as ground truth. The results of both algorithms are presented in the following sections.

4.3.1 Daytime Results

During daytime both algorithms were executed for the five heavy storms selected from 2003 to 2007. To perform a fair comparison, both algorithms were run with the same observations. Rainfall detection information was obtained from NEXRAD and used as ground truth to obtain the metrics of hit rate (HIT), probability of detection (POD), false alarm (FAR), bias (BIAS) and performance index (INDEX). Table 4.5 present the results obtained for the cloud classification system. The HE results are presented on table 4.6. From the performance results it can be seen that the cloud classification obtained better results than the HE on every storm. The cloud classification system obtained an average of 0.87, 0.47, 0.40, 0.78, and 0.35 in HIT, POD, FAR, BIAS, and INDEX respectively. The HE obtained an average of 0.77, 0.38, 0.67, 1.17, and 0.50 in HIT, POD, FAR, BIAS, and INDEX respectively. It can be seen from the results that the cloud classification system reduced the false alarm while maintaining a good hit rate and a reasonable probability of detection when compared with the HE.

TABLE 4.5 Cloud classification system performance during daytime.

Storm Date	**HIT**	**POD**	**FAR**	**BIAS**	**INDEX**
Nov-03	0.79	0.78	0.15	0.92	**0.19**
Dec-03	0.89	0.30	0.75	1.18	**0.52**
Apr-05	0.98	0.38	0.49	0.74	**0.38**
May-05	0.89	0.07	0.44	0.13	**0.49**
Oct-07	0.79	0.80	0.15	0.93	**0.19**
Average	**0.87**	**0.47**	**0.40**	**0.78**	**0.35**

TABLE 4.6 Hydro-Estimator performance during daytime.

Storm Date	**HIT**	**POD**	**FAR**	**BIAS**	**INDEX**
Nov-03	0.63	0.64	0.31	0.93	**0.34**
Dec-03	0.78	0.42	0.86	2.95	**0.55**
Apr-05	0.97	0.002	0.99	0.14	**0.67**
May-05	0.83	0.07	0.89	0.68	**0.66**
Oct-07	0.64	0.78	0.31	1.13	**0.30**
Average	**0.77**	**0.38**	**0.67**	**1.17**	**0.50**

4.3.2 Nighttime Results

During nighttime equal to daytime both algorithms were executed for the five heavy storms selected from 2003 to 2007. Rainfall detection information was obtained from NEXRAD and used as ground truth to obtain the metrics of hit rate (HIT), probability of detection (POD), false alarm (FAR), bias (BIAS) and performance index (INDEX). Table 4.7 present the results obtained for the cloud classification system. The

HE results are presented on table 4.8. From the performance results it can be seen that the cloud classification obtained better results than the HE on every storm. The cloud classification system obtained an average of 0.79, 0.61, 0.35, 0.97, and 0.31 in HIT, POD, FAR, BIAS, and INDEX respectively. The HE obtained an average of 0.69, 0.66, 0.49, 1.37, and 0.38 in HIT, POD, FAR, BIAS, and INDEX respectively. Similar to the daytime results the cloud classification system reduced the false alarm while maintaining a good hit rate and a reasonable probability of detection compared with the HE.

TABLE 4.7 Cloud classification system performance during nighttime.

Storm Date	**HIT**	**POD**	**FAR**	**BIAS**	**INDEX**
Nov-03	0.78	0.63	0.32	0.92	**0.30**
Dec-03	0.79	0.61	0.53	1.28	**0.38**
Apr-05	0.96	0.52	0.39	0.85	**0.30**
May-05	0.72	0.58	0.21	0.73	**0.30**
Oct-07	0.72	0.73	0.32	1.08	**0.29**
Average	**0.79**	**0.61**	**0.35**	**0.97**	**0.31**

.

TABLE 4.8 Hydro-Estimator performance during nighttime.

Storm Date	**HIT**	**POD**	**FAR**	**BIAS**	**INDEX**
Nov-03	0.65	0.57	0.52	1.20	**0.43**
Dec-03	0.68	0.76	0.64	2.15	**0.40**
Apr-05	0.96	0.47	0.38	0.76	**0.31**
May-05	0.57	0.66	0.46	1.23	**0.41**
Oct-07	0.61	0.83	0.45	1.52	**0.34**
Average	**0.69**	**0.66**	**0.49**	**1.37**	**0.38**

4.4 Rainy Warm Clouds Detection

One of the objectives of this study and maybe the most important of all is to improve rainfall detection from warm clouds. To measure this improvement the last performance measurement obtained from both of the algorithms is the percentage of warm cloud detection. Defined as the ratio of pixels that produced rainfall over 235K detected by the algorithms to those detected by NEXRAD expressed as a percentage. Similar to all the measurements previously obtained the warm cloud detection performance was obtained during daytime and nighttime. From table 4.9 it can be seen that the cloud classification system (CCS) detected more rainfall produced by warm clouds than the Hydro-Estimator (HE) in 4 out of the 5 storms studied. In average the CCS detected 39% of the rainfall produced by warm clouds and the HE 28%. During nighttime the CCS detected more rainfall produced by warm clouds than the HE in every storm. In average during nighttime the CCS detected 44% of the rainfall produced by warm clouds and the HE 25%. The results obtained indicate that using a cloud classification system with multiple variables as input does improve the warm cloud rainfall detection.

TABLE 4.9 Rainy warm clouds detection performances during daytime.

Storm Date	**CCS**	**HE**
Nov-03	45%	31%
Dec-03	35%	13%
Apr-05	30%	1%
May-05	6%	19%
Oct-07	80%	74%
Average	**39%**	**28%**

TABLE 4.10 Rainy warm clouds detection performances during nighttime.

Storm Date	**CCS**	**HE**
Nov-03	40%	31%
Dec-03	35%	18%
Apr-05	58%	31%
May-05	34%	17%
Oct-07	55%	27%
Average	**44%**	**25%**

4.5 Graphical Comparison

In order to perform a visual comparison between the cloud classification system, the HE and NEXRAD in terms of rainfall detection, an image was randomly selected. From figure 4.1 it can be seen that the HE overestimates rainfall detection; whereas, the cloud classification system exhibits detection that resembles NEXRAD. This agrees with the results obtained on the discrete evaluation where the HE exhibits a high FAR.

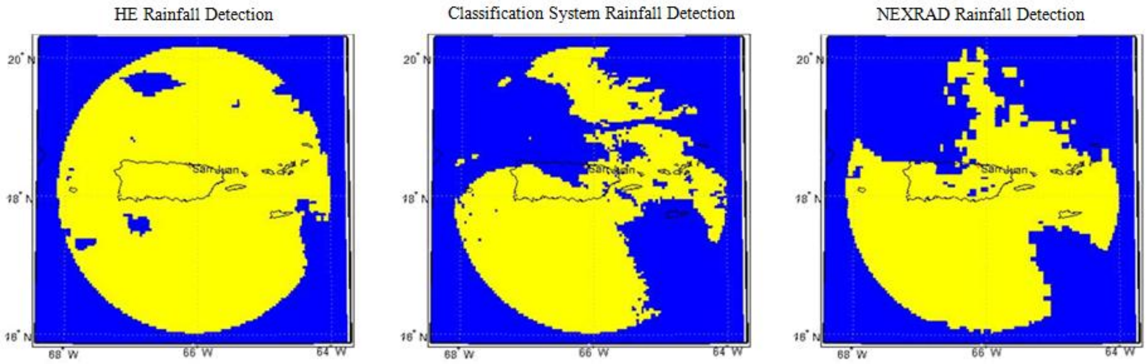

HE Rainfall Detection
Classification System Rainfall Detection
NEXRAD Rainfall Detection
20° N
18° N
16° N
68° W
66° W
64° W
San Juan

Figure 4.1 Rainfall detection comparison between the HE (left), the cloud classification system (center), and NEXRAD (right).

4.6 Study Limitations

Some limitations this study had are:

- It was not possible to obtain the effective radius and therefore no direct microphysical measurements were used as input for the classification system.
- NEXRAD information was assumed to be ground truth for the classification system training (i.e., feature selection process and centroid selection process).

- Both the cloud classification algorithm and the Hydro-estimator were validated using NEXRAD information as ground truth.
- NEXRAD information even though is assumed to be ground truth is not perfect.
- Weather prediction model files (NAM files) needed to run the Hydro-estimator are made available through email request. This made it difficult to add short duration rainfall events to this study.

5 Conclusions and Future Work

5.1 Conclusions

The objectives of this study were: 1) to create a cloud classification system to improve rainfall detection; 2) to identify remotely sensed variables that may improve detection of rainfall, especially rainfall produced by clouds with brightness temperature over 235K; 3) to develop a rainfall detection algorithm, which combines a cloud classification system and multiple remotely sensed variables obtained from geostationary satellite; 4) to validate the performance of the algorithm using NEXRAD data as ground truth.

Maximum likelihood, a supervised classification method used in remote sensing practices, was selected as the classification method to be implemented. This method provides the capability of classifying multiple variables into previously defined classes. Given that from GOES multiple remotely sensed features can be extracted Maximum Likelihood fits well when identifying, which of these features defines best the occurrence of rainfall over Puerto Rico (PR).

The selected classification method was used on this study to identify, which remotely sensed features combined define best the occurrence of rainfall in PR. It was determined that during daytime the features that best describe rainfall occurrence are the combination of Visible Reflectance centered at 0.65µm, brightness temperature difference between channels 2 (3.9µm) and 4 (10.7µm), brightness temperature from channel 4 (10.7µm) and Albedo centered at 3.9µm. During nighttime when visible reflectance and albedo are not available, the remotely sensed that best describe rainfall occurrence were the combination of the brightness temperature difference between channels 2 (3.9µm) and 4 (10.7µm) and the brightness temperature difference between channels 3 (6.9µm) and 4 (10.7µm).

Validations were conducted with the cloud classification system using the centroids set obtained from the storm of October 2007 for daytime and with those obtained from May 2005 for the nighttime. Detection performance was measured in terms of hit rate (HIT), probability of detection (POD), false alarm rate (FAR), bias and a performance index, which combines the HIT, FAR and POD to measure the overall performance. The metrics were obtained for both the cloud classification system developed and the Hydro-Estimator. Results show that the classification system performed better during daytime as well as at nighttime obtaining an average of 0.87, 0.47, 0.40, 0.78, and 0.35 in HIT, POD, FAR, BIAS, and INDEX respectively while the HE obtained an average of 0.77, 0.38, 0.67, 1.17, and 0.50 during daytime. During nighttime the cloud classification system obtained an average of 0.79, 0.61, 0.35, 0.97, and 0.31 in HIT, POD, FAR, BIAS, and INDEX respectively while HE obtained an average of 0.69, 0.66, 0.49, 1.37, and 0.38.

In terms of improvements achieved detecting rainfall produced by clouds with warm tops (i.e., brightness temperature above 235K), the cloud classification system obtained better results than the HE during daytime and nighttime with the exception of one storm during daytime. During daytime the cloud classification system detected in average 11% more rainfall produced by warm clouds than the HE and 19% more during nighttime.

In conclusion the developed algorithm obtained better results than those obtained by the HE on the storms studied. It improves overall detection of rainfall as well as rainfall detection produced by warm clouds over Puerto Rico. This implies that using multiple remotely sensed features have more potential than just relying on brightness temperature from a single channel when detecting rainfall over Puerto Rico. The developed algorithm is a promising rainfall detection technique, which can be used in conjunction with a rainfall estimation algorithm to derive rain rates for Puerto Rico.

5.2 Recommendations and Future Work

Some recommendations for future research works are:

- Validate the detection performance of the cloud classification system against rain gauges at an island scale.

- Validate the detection performance of the cloud classification system against the rain gauge network located on western Puerto Rico.

- Validate the detection performance of the cloud classification system against radars located on western Puerto Rico (e.g., CASA radars and proposed TropiNet radars).

- Develop a method or the look-up tables necessary to obtain the effective radius of cloud particles for warm clouds.

- Define relationships between the features that best detect rainfall and rain rates for Puerto Rico.

- Combine the cloud classification system with a rainfall estimation algorithm to improve rainfall estimation for Puerto Rico.

- Extend the cloud classification system to include the continental U.S as well as other subtropical regions.

- Implement a real time operational version of the cloud classification algorithm.

References

Alder, R. F., and Negri, A. J., "A Satellite Infrared Technique to Estimate Tropical Convective and Stratiform Rainfall", Journal of Applied Meteorology, vol. 27, pp. 30-51, 1888.

American Meteorological Society (AMS), "Glossary of Metrology", http://amsglossary.allenpress.com/glossary/search?id=brightness-temperature1 , accessed: March 2011.

Arkin, P. A., and Meisner, B. N., "The Relationship Between Large Scale Convective Rainfall and Cold Cloud Over the Western Hemisphere During 1982–84", *Monthly Weather Review*, vol. 115, pp. 51-74, 1987.

Ba, M. B., and Gruber, A., "GOES Multispectral Rainfall Algorithm (GMSRA)", *Journal of Applied Meteorology*, vol. 40, pp. 1500-1514, 2000.

Biswas A. K., "The automatic Rain-Gauge of Sir Christopher Wren, F.R.S.", *Notes and Records of The Royal Society of London*., vol. 22 no. 1 94-104, 1967.

Citizen Weather Observer Program (CWOP), "Site Information for TJUA in San Juan/cayey, US", http://weather.gladstonefamily.net/site/TJUA , last modified: 15 May 2011

Cruz-González, B., "Validación del Algoritmo Hidro-Estimador en la Region de Puerto Rico", *MS thesis, Industrial Engineering Department, University of Puerto Rico, Mayagüez, Campus*, 2006.

Ferraro, R. R., "Special Sensor Microwave Imager Derived Global Estimates for Climatological Applications", *Journal of Atmospheric and Oceanic Technology*, vol. 12, pp. 5-19, 1997.

Ferraro, R. R., and Marks, G. F., "The Development of SSM/I Rain-Rate Retrieval Algorithms Using Ground-Based Radar Measurements", *Journal of Atmospheric and Oceanic Technology*, vol. 12, pp.755-770, 1995.

Grossberg, S., “Embedding Fields: A Theory of Learning with Physiological Implications”, *Journal of Mathematical Physics*, vol. 6, pp. 209-239, 1969.

Habib, E., Meselhe, E. A., and Aduvala, A.. “Effect of Local Errors of Tipping-Bucket Rain Gauges on Rainfall-Runoff Simulations” *ASCE Journal of Hydrologic Engineering*, vol. 13(6), pp. 411-528, 2008.

Harmsen, E. W., Gomez Mesa, S. E., Cabassa, E., Ramirez Beltran, N. D., Pol, S. C., Kuligowski R. J., and Vasquez, R., “Satellite Sub-Pixel Rainfall Variability”, *WSEAS Transactions on Signal Processing*, Issue 8, vol. 7, pp. 504-513, 2008.

Hsu, K., Gao, X., Sorooshian, S., and Gupta, H. V., “Precipitation Estimation from Remotely Sensed Information Using Artificial Neural Networks”, *Journal of Applied Meteorology*, vol. 36, pp. 1176-1190, 1997.

Hsu, H., Gupta, V., Gao, X., and Sorooshian, S., “Estimation of Physical Variables from Multichannel Remotely Sensed Imagery Using a Neural Network: Application to Rainfall Estimation”, *Water Resources Research*, vol. 35, pp. 1605-1618, 1999.

Hong, Y., Hsu, K. L., Sorooshian, S., and Gao, X. G., “Precipitation Estimation from Remotely Sensed Imagery Using an Artificial Neural Network Cloud Classification System”, *Journal of Applied Meteorology*, vol. 43, no. 12, pp. 1834-1852, 2004

Jung, H-S., Lim, G-H., and Oh, J-H., 2001, “Interpretation of the Transient Variations in the Time Series of Precipitation Amounts in Seoul, Korea. Part I: Diurnal Variation”, *Journal of Climate*, vol. 14, pp. 2989-3004, 2001.

Kelleher, K.E., Droegemeier, K.K. , Levit, J. J., Sinclair, C., Jahn, D. E., Hill, S. D., Mueller, L., Qualley, G., Crum, T. D., Smith, S. D., Del Greco, S. A., Lakshmivarahan, S., Miller, L., Ramamurthy, M., Domenic, B., and Fulker, D. W., "A real-time delivery system for NEXRAD, Level II data via the internet”, *Bulletin of the American Meteorological Society*, vol. 88, 1045-1057, 2007.

Kohonen, T., “Self-Organized Formation of Topologically Correct Feature Maps”, *Biological Cybernetics*, vol. 43, pp. 59-69, 1982.

Kuligowski, R. J., “A self-calibrating GOES rainfall algorithm for short-term rainfall estimates”, *Journal of Hydrometeorology*, vol. 3, pp. 112-130, 2002.

Lindsey, D., and Grasso, L., “An Effective Radius for Thick Ice Clouds Using GOES”, *Journal of Applied Meteorology and Climatology*, vol. 47, pp. 1222-1231, 2008.

Nakajima, T., and King, M. D., “Determination of the Optical Thickness and Effective Particle Radius of Clouds from Reflected Solar Radiation Measurements. Part I: Theory”, *Journal of Atmospheric Sciences*, vol. 47, pp. 1878-1893, 1990.

NOAA/NESDIS Office of Satellites Operations (OSO), “Geostationary Satellites”, http://www.oso.noaa.gov/goes/ , accessed: April 2011.

NOAA/NESDIS Satellite and Information Services (NOAASIS), “GOES Imager Instrument”, http://noaasis.noaa.gov/NOAASIS/ml/imager.html , accessed: April 2011a.

NOAA/NESDIS Satellite and Information Services (NOAASIS), “GOES 13 Spacecraft Status Summary”, http://www.oso.noaa.gov/goesstatus/spacecraftStatusSummary.asp?spacecraft=13 , accessed: April 2011b.

NCDC, “National Climatic Data Center: Data Documentation for DSI – 7000”, *NEXRAD Level III*, Asheville, NC, 2005.

Ramírez-Beltrán, N. D, Kuligowski, R. J., Harmsen, E. W., Castro, J. M., Cruz-Pol, S., and Cardona-Soto, M. J., “Rainfall Estimation from Convective Storms Using the Hydro-Estimator and NEXRAD”, *WSEAS Transaction on Systems*, vol. 7, no. 10, pp. 1016-1027, 2008.

Ramírez-Beltran, N.D, Kuligowski, R.J., Castro, J.M.,Cardona-Soto, M, and Vasquez, R., “A Projection Algorithm for Satellite Rainfall Detection”, *WSEAS Transaction on Systems*. Issue 6, vol 8, pp 763-772, 2009.

Rosenfeld, D., and Gutman, G., “Retrieving Microphysical Properties Near the Tops of Potencial Rain Clouds by Multi Spectral Analysis of AVHRR Data”, *Atmospheric Research*, vol. 34, pp. 259-283.

Scofield R. A. and Kuligowski R. J., “Status and Outlook of Operational Satellite Precipitation Algorithms for Extreme-Precipitation Events”, *Weather and Forecasting*, vol. 18, no. 6, pp. 1037-105, 2003.

Schott, J. R., “Remote Sensing: The Image Chain Approach”, Second Edition, *Oxford University Press, NY*, 2007.

Sorooshian, S., Hsu, K., Hong, Y., “Global Precipitation Estimation from Satellite Image Using Artificial Neural Networks”, *UNESCO science meeting on Water And Development Information for Arid Lands-A Global Network*, 2005.

Sorooshian, S., Hsu, K.-L., Gao, X., Gupta, H. V., Imam, B., and Braithwaite, D., “An Evaluation of PERSIANN System Satellite-Based Estimates of Tropical Rainfall”, *Bulletin of the American Meteorological Society*, vol. 81, pp. 2035-2046, 2000.

Tjemkes, S. A., van de Berg, L., and Schemetz J., “Warm Water Vapor Pixels Over High Clouds as Observed by Meteosat”, *Beitraege zur Physik der Atmosphaere*, vol 70. pp. 15-21, 1997.

U.S. Geological Survey, “National Water Information System Data Available on the World Wide Web (Water Data for the Nation)”, http://waterdata.usgs.gov/nwis/, accessed: July, 2011

Vicente, G., Scofield, R. A., and Mensel, W. P., “The Operational GOES Infrared Rainfall Estimation Technique”, *Bulletin of the American Meteorological Society*, vol. 79, pp. 1881-1898, 1998.

Vila D. and Velasco I., “Some Experiences On Satellite Rainfall Estimation Over South America”, *1st IPWG Workshop EUMETSAT SAF NWC, Instituto Nacional de Meteorología, Madrid, Spain*, 23-27 September 2002.

Weng, F., Ferraro, R. R., and Grody, N. C., “Global Precipitation Estimation Using Defense Meteorological Satellite Program F10 and F11 Special Sensor Microwave Imager Data”, *Journal of Geophysical Research*, vol. 99, pp. 14 493- 14 502, 1994.

Wu, X., and Sun F., “Post-launch calibration of GOES Imager visible channel using MODIS”, Proceedings of SPIE, 2005.

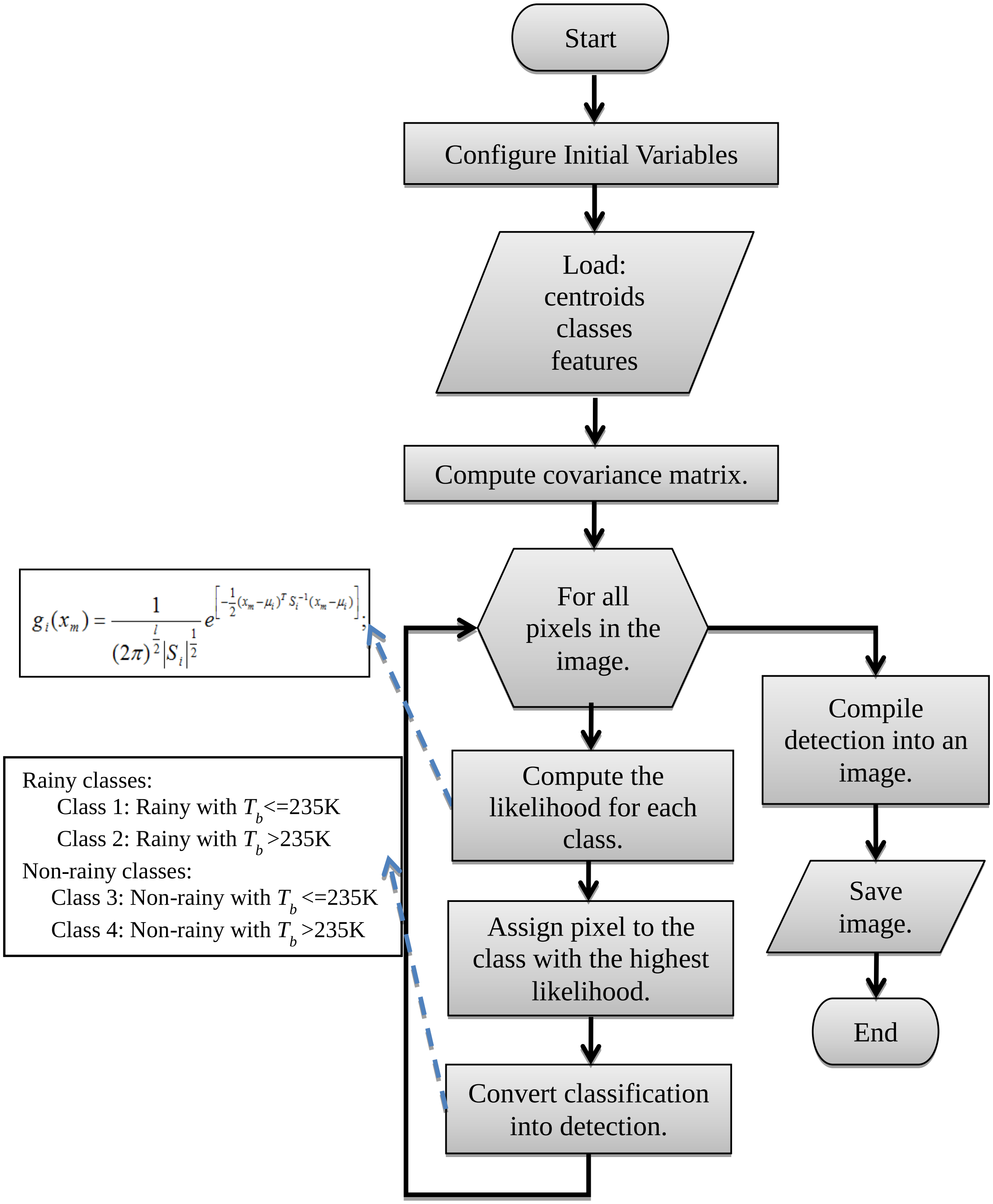
APPENDIX A CLOUD CLASSIFICATION SYSTEM MATLAB CODE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The following matlab code classifies GOES images using the
% maximum likelihood function, which returns classification and detection
% and saves them in a MATLAB file (.mat).
%
% inputs:  - path of directory containing data files
%          - all data files arranged in cell arrays:
%             All_VisRef -> Visible Reflectance
%             All_Tb2    -> Brightness Temperature for GOES channel 2
%             All_Tb3    -> Brightness Temperature for GOES channel 3
%             All_Tb4    -> Brightness Temperature for GOES channel 4
%             All_Alb    -> Albedo
%             Time       -> UTC times
%             Latm       -> Latitudes
%             Lonm       -> Longitudes
%          - path of directory containing the centroids files
%          - centroids file, which contains the centroids and the classified
%            data used to obtain the centroids
% outputs: - MATLAB file containing all rainfall detection provided by the
%            cloud classification system and the cloud classification
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; %Clears all information from workspace
clc; %Clears command window
close all; %Close all windows and figures

%month and year of the storm to be evaluated
monthYear = 'May2005';

%path where the data file is saved
filePath=('E:\Investigacion\MatlabApplications\DataSets\GOES-12\' monthYear
'\StormData.mat');

dayTime = 'day'; %set to 'day' or night depending on the period to evaluate

%set the centroid file path depending on the period of time to be evaluated
%and the features that will be used
if(dayTime=='day')
    centroidsFilePath=('E:\Investigacion\MatlabApplications\DataSets\GOES-
12\Centroids\Oct2007.mat');
    %[VisRef,Tb2-4,Tb4,Alb]
    features = [1245];
else
    cetroidsFilePath=('E:\Investigacion\MatlabApplications\DataSets\GOES-
12\Centroids\May2005.mat');
    %[Tb2-3,Tb3-4]
    features = [45];
end

%sets image resolution
res=4/deg2km(1);
```

```
%latitude and longitude limits of image
latlim=[15.89 20.3];
lonlim=[-68.43 -63.73];

%computes the covariances matrices
covariances =
{cov(classes{1}(:,features));cov(classes{2}(:,features));cov(classes{3}(:,featu
res));cov(classes{4}(:,features))};

%iterates through all the items in the cell arrays and obtains cloud
%classification and rainfall detection with the maximum likelihood method
for i=1:length(Time)
    %- depending on the daytime the information sent to the function
    %  maximumLikelihood is different since the centroids during daytime are
not
    %  equal to those used during nightime
    %- the function maximumLikelihood returns the classification and the
    %  detection for the entire images
    if(dayTime=='day')
        [classification,detection] =
maximumLikelihood(All_VisRef{i},All_Tb2{i}-All_Tb4{i},All_Tb3{i}-
All_Tb4{i},All_Tb4{i},All_Alb{i},centroids,covariances,features);
    else
        [classification,detection] =
maximumLikelihood(All_Tb2{i},All_Tb3{i},All_Tb4{i},All_Tb2{i}-
All_Tb4{i},All_Tb3{i}-All_Tb4{i},centroids,covariances,features);
    end
    All_Classification{i,1} = classification;
    All_Detection{i,1} = detection;
end

%saves classifications, detections and identification data into a .mat file
save(['E:\Investigacion\MatlabApplications\DataSets\GOES-12\' monthYear
'\MALClassification.mat'],'monthYear','Time','All_Classification','All_Detectio
n','Latm','Lonm')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The following matlab code implements the Maximum Likelihood
% classification method. It classifies all the pixels in an image into
% 4 previously defined classes and translate the classification into
% rainfall detection.
% Inputs: - features used for the classification that will take place
%         - centroids of the classes in, which pixels will be classified
%         - classes' covariances
%         - list of features that will be used for the classification
% Outputs: - Maximum Likelihood classification
%          - Classification translated into rainfall detection
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function
[classification,detection]=maximumLikelihood(feat1,feat2,feat3,feat4,feat5,cent
roids,covariances,features)

classification = zeros(size(feat1));
detection = zeros(size(feat1));
```

```
filcol = size(feat1);

for m = 1:filcol(1)
    for n = 1:filcol(2)
       X = [feat1(m,n) feat2(m,n) feat3(m,n) feat4(m,n) feat5(m,n)];
       X = X(features);
       l = length(features);
       likelihood = [];
       dist = [];
       for i = 1:4
           mahalDist = [(X-centroids{i}(:,features))*inv(covariances{i})*(X-
centroids{i}(:,features))'];
           likelihood =
[likelihood;((1/((2*pi)^(l/2)*sqrt(det(covariances{i}))))*exp(-
1/2*mahalDist))];
       end

       temp = max(likelihood);

       if(isnan(temp))
           classification(m,n) = NaN;
       else
           tempClass = find(likelihood == temp);

           if(length(tempClass)>1)
               classification(m,n) = NaN;
           else
               classification(m,n) = tempClass;
           end
       end
    end
end
detection(find(classification ==1|classification ==2)) = 1;
detection(find(classification ==3|classification ==4)) = 0;
```

APPENDIX B CLOUD CLASSIFICATION SYSTEM FLOWCHART


Start
Configure Initial Variables
Load:
centroids
classes
features
Compute covariance matrix.
For all pixels in the image.
Compute the likelihood for each class.
Assign pixel to the class with the highest likelihood.
Convert classification into detection.
Compile detection into an image.
Save image.
End
$g_i(x_m) = \frac{1}{(2\pi)^{\frac{l}{2}}|S_i|^{\frac{1}{2}}} e^{\left[-\frac{1}{2}(x_m-\mu_i)^T S_i^{-1}(x_m-\mu_i)\right]};$
Rainy classes:
Class 1: Rainy with T_b<=235K
Class 2: Rainy with T_b>235K
Non-rainy classes:
Class 3: Non-rainy with T_b <=235K
Class 4: Non-rainy with T_b >235K

High level flowchart of the cloud classification system.

APPENDIX C VISIBLE REFLECTANCE MATLAB CODE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The following matlab code converts all NetCDF files (.nc)
% in a directory from GOES-12 counts (band 1) into visible reflectance
% and saves them in MATLAB files (.mat) files.
%
% inputs:  - path of directory containing NetCDF files
%          - NetCDF files
% outputs: - MATLAB files containing visible reflectance
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; %Clears all information from workspace
clc; %Clears command window
close all; %Close all windows and figures

%month and year of the storm to be evaluated
monthYear = 'Oct2007';

%directory path where NetCDF are saved
directory=['E:\Investigacion\MatlabApplications\DataSets\GOES-12\' monthYear
'\NC\BAND_01\'];

%finds all NetCDF files on the directory specified
toRead=dir([directory '.nc*']);

%sets image resolution
res=4/deg2km(1);

%latitude and longitude limits of image
latlim=[15.89 20.3];
lonlim=[-68.43 -63.73];

for filesCounter = 210:length(toRead)

    %displays name of file to be used
    disp(toRead(filesCounter).name);
    %opens NetCDF file and loads it into memory
    nc=netcdf([directory toRead(filesCounter).name],'write');

    %retrieves GOES counts, latitudes and longitudes from NetCDF file
    counts=nc{'data'}(:);
    La=nc{'latitude'}(:);
    Lo=nc{'longitude'}(:);
    clear nc;

    %builds latitudes and longitudes matrixs from specified limits
    %at a 4km*4km resolution
    [Lonm,Latm]=meshgrid(lonlim(1):res:lonlim(2),latlim(2):-res:latlim(1));

    %reference GOES counts so that they can be visualized as images and
    %mapped
    X = griddata(Lon,Lat,counts,Lonm,Latm);
    X = X/32;%data type change from 16bit to 10 bit
    %sets k and C
    %k value can be found in:

%http://www.star.nesdis.noaa.gov/smcd/spb/fwu/homepage/GOES_Imager_Vis_PreCal.p
hp
    %C value can be found in:
    %GOES_12:

%http://www.star.nesdis.noaa.gov/smcd/spb/fwu/homepage/GOES_Imager_Vis_OpCal_G1
2.php
    %GOES-13:

%http://www.star.nesdis.noaa.gov/smcd/spb/fwu/homepage/GOES_Imager_Vis_OpCal_G1
3.php
    k = 0.001141; %Goes-12
    %k = 0.001160; %goes-13

    C = 1.355; %Oct2007

    Xspace = 29;

    %visible reflectance is computed for pre-launched calibration
    Apre = k*(X-Xspace);

    %visible reflectance is corrected for post-launched calibration
    VisRef = Apre*C;

    VisRef = flipud(VisRef);
    Latm = flipud(Latm);

    %saves visible reflectance, latitudes and longitudes into a .mat file
    save([directory
toRead(filesCounter).name(1:length(toRead(filesCounter).name)-3)
'.VisRef.mat'],'VisRef','Latm','Lonm')

end
```

APPENDIX D BRIGHTNESS TEMPERATURE MATLAB CODE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The following matlab code converts all NetCDF files (.nc)
% in a directory from GOES counts into brightness temperatures
% and saves them in MATLAB files (.mat) files.
%
% inputs:  - path of directory containing NetCDF files
%          - NetCDF files
%
% outputs: - MATLAB files containing brightness temperatures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; %Clears all information from workspace
clc; %Clears command window
close all; %Close all windows and figures

%directory path where NetCDF are saved
directory=('E:\Investigacion\MatlabApplications\DataSets\GOES-12\Oct2007\');

%sets image resolution
res=4/deg2km(1);

%latitude and longitude limits of image
latlim=[15.89 20.3];
lonlim=[-68.43 -63.73];

%finds all NetCDF files on the directory specified
toRead=dir([directory '.nc*']);

for filesCounter=1:length(toRead)
    %displays name of file to be used
    disp(toRead(filesCounter).name);
    %opens NetCDF file and loads it into memory
    nc=netcdf([direc toRead(filesCounter).name],'write');

    %retrieves GOES counts, latitudes and longitudes from NetCDF file
    counts=nc{'data'}(:);
    La=nc{'latitude'}(:);
    Lo=nc{'longitude'}(:);

    %builds latitudes and longitudes matrixes from specified limits
    %at a 4km*4km resolution
    [Lonm,Latm]=meshgrid(lonlim(1):res:lonlim(2),latlim(2):-res:latlim(1));

    %reference GOES counts so that they can be visualized as images and
    %mapped
    countsM = griddata(Lo,La,counts,Lonm,Latm);

    %converts GOES counts into brightness temperatures
    Tb=zeros(size(countsM));
    temp1=find(countsM<176);
    temp2=find(countsM>=176);
    Tb(temp1)=330-0.5*countsM(temp1);
    Tb(temp2)=418-countsM(temp2);
    Tb = flipud(Tb);
    Latm = flipud(Latm);

    %saves brightness temperatures, latitudes and longitudes into a .mat file
    save([directory '\'
toRead(filesCounter).name(1:length(toRead(filesCounter).name)-3)
'.Tb.mat'],'Tb','Latm','Lonm')

end
```

APPENDIX E ALBEDO MATLAB CODE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The following matlab obtains albedo centered at 3.9um
% using as input the brightness temperatures from bands 2 and 4
% of GOES and saves the outputs in MATLAB files (.mat) files.
%
% inputs:  - path of directory containing the brigtness temperature files
%          - brightness temperatures files (.mat)
% outputs: - MATLAB files containing albedo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This algorithm is a modified translation of Dr. Daniel Lindey (CIRA)original
% fortran algorithm [effrad.pgm and necesary functions (geo_parms.f90 &
lookup.for)].
% by: Melvin J. Cardona-Soto (NOAA-CREST UPRM)
% The original version obtained the effective radius whiles this version
% only obtains the albedo.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; %Clears all information from workspace
clc; %Clears command window
close all; %Close all windows and figures

%month and year of the storm to be evaluated
monthYear = 'Sept2008';

%sets identifications of satelites
iSat = 12; %12 or 13
idenSat = 78; %78 for GOES-12, 180 for GOES-13

%directory path where brightness temperature files are saved
directory = [cd '\' monthYear '\Tb\'];

%finds all NetCDF files on the directory specified
toRead2 = dir([directory 'BAND_02\' '*BAND_02.Tb.mat*']);
toRead4 = dir([directory 'BAND_04\' '*BAND_04.Tb.mat*']);

for fileIndex = 1:size(toRead2)
    %the following lines extracts the brigtnes tempertarues on the .mat
    %files and saves them in the variables Tb2 and Tb4
    disp([toRead2(fileIndex).name]);
    load([directory 'BAND_02\' toRead2(fileIndex).name])
    Tb2 = Tb;
    clear Tb;
    disp([toRead4(fileIndex).name]);
    load([directory 'BAND_04\' toRead4(fileIndex).name])
    Tb4 = Tb;
    clear Tb;

    %sets image resolution
    res=4/deg2km(1);
    %latitude and longitude limits of image
    latlim=[15.89 20.3];
    lonlim=[-68.43 -63.73];

    %obtains wavelenght and wavenumber for each band (2 and 4)
    [waveLenght2,waveNumber2]=sat2wave(idenSat,2);
    [waveLenght4,waveNumber4]=sat2wave(idenSat,4);

    %obtains the julian day and utc time from file name
    jday = str2num(toRead2(fileIndex).name(13:15));
    utc = round(str2num(toRead2(fileIndex).name(17:22))/100);


    solarZenithAngle = zeros(size(Tb2));
    albedo = zeros(size(Tb2));

    %iterates on every pixel on the image to get albedo for each one
    Tb2Size = size(Tb2);
    for r=1:Tb2Size(1)
        for c=1:Tb2Size(2)

            %obtains solar zenith angle
            [solarZenithAngle(r,c)]=
getSolarAZAngles(Latm(r,c),Lonm(r,c),jday,utc);

            %calculates radiances
            Rad(r,c) = planck(waveNumber2,Tb2(r,c));
            Rad_em(r,c) = planck(waveNumber2,Tb4(r,c));

            %calculates the albedo
            sunIrradiance = 4.86793*cosd(solarZenithAngle(r,c));
            albedo(r,c) = ((Rad(r,c)-Rad_em(r,c))/(sunIrradiance-
Rad_em(r,c)))*100;

        end %end for r
    end %end for c

    Alb = albedo;

save([cd '\' monthYear '\NC\VisRef\'
toRead4(fileIndex).name(1:length(toRead4(fileIndex).name)-14)
'Alb.mat'],'Alb','Latm','Lonm')

end %end main for


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The following matlab code calculates the wavelenth and wavenumber for a
% given satellite or instrument and band
% Inputs: - setellite or sensor id
%         - band number
% Outputs: - wavelengh
%          - wavenumber
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [wavel,waven] = sat2wave(iden,kband)

      wavel=0.;
      if((iden>=70)&(iden<=79)&(mod(iden,2)==0)|(iden==180))

% GOES Imager

         if(kband==2)
             wavel=3.9;
         elseif(kband==4)
             wavel=10.7;
         elseif(kband==5)
             wavel=12.0
         end
      end

      if(wavel==0)
          disp('- satellite/sensor or band not available -');
      end

% wavenumber (cm-1) from wavelength (um)

      waven=1.e4/wavel;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The following matlab code calculates the solar zenith angle using the
% coordinates of the actual pixel
% Inputs: - latitude
%         - longitude
%         - utc time
%         - julian day
% Outputs: - solar zenith angle
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [solarZenithAngle]= getSolarAZAngles(lat,lon,utc,jday)

[equationOfTime, sunDeclnAngle] = sunTable(jday);

[TrueSolarTime] = solarTime(utc,lon,equationOfTime);

SolarHourAngle=(TrueSolarTime-12)*15;

% calculate the cosine of the zenith angle
ZenithAngleCos = sind(lat)*sind(sunDeclnAngle) +
cosd(lat)*cosd(sunDeclnAngle)*cosd(SolarHourAngle);

if(ZenithAngleCos > 1.0)
    ZenithAngleCos=1.0;
end

if(ZenithAngleCos < -1.0)
    ZenithAngleCos = -1.0;
end

% calculate zenith
solarZenithAngle = acosd(ZenithAngleCos) ;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The following matlab code calculates the solar zenith angle using the
% julian day
% Inputs: - julian day
% Outputs: - equation of time
%          - sun declination angle
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [equationOfTime, sunDeclnAngle] = sunTable (jday)

beta = (360*(jday-81))/364;
equationOfTime = 9.87*sind(2*beta) - 7.53*cosd(beta) -1.5*sind(beta);

sunDeclnAngle = 23.45 * sind(360 *((284+jday)/365));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The following matlab code calculates the true solar time using the utc,
% longitude and the equation of time
% Inputs: - utc time
%         - longitude
%         - equation of time
% Outputs: - true solar time
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [TrueSolarTime] = solarTime(utc,Lon,equationOfTime)

% UTC to local time
LonHR = floor(Lon/15);
LocalTime = utc + (LonHR*100);

% Local time to minutes
Hours = (floor(LocalTime/100)*60);
Left = (LocalTime - floor(LocalTime/100)*100);
Mins = Hours+Left;

% Calculate Zone Correction
LongMn = (Lon/15-floor(Lon/15))*15;

% Zone and equation of time correction
Lmins = Mins+equationOfTime+4*LongMn;
TrueSolarTime=Lmins/60.0;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The following matlab code implements the plank function to calculate the
% radiance using the wavenumner and temperature
% Inputs: - wavenumber
%         - temperature
% Outputs: - radiance
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [radiance] = planck(waveNumber,temperature)

% Constants
c1 = 1.1909e-5;
c2 = 1.438;

radiance = c1*waveNumber^3/(exp(c2*waveNumber/temperature)-1);
```