Optimal Wavelet Features for an Infrared Satellite Precipitation Estimate Algorithm

Majid Mahrooghy[1, 2], Valentine G. Anantharaj[2], Nicolas H. Younan[1, 2], and James Aanstoos[2]
[1]Department of Electrical Engineering, Mississippi State University, Mississippi
[2]Geosystems Research Institute, Mississippi State University, Mississippi

***Abstract*-A satellite precipitation estimation algorithm based on wavelet features is investigated to find the optimal wavelet features in terms of wavelet family and sliding window size. In this work, the infrared satellite based images along with ground gauge (radar corrected) observations are used for the retrieval rainfall. The goal of this work is to find an optimal wavelet transform to represent better features for cloud classification and rainfall estimation. Our approach involves the following four steps: 1) segmentation of infrared cloud images into patches; 2) feature extraction using a wavelet-based method; 3) clustering and classification of cloud patches using neural network, and 4) dynamic application of brightness temperature (Tb) and rain rate relationships, derived using satellite observations. The results show that Haar and Symlet wavelets with sliding window size 5×5 have better estimate performance than other wavelet families and window sizes.**

***Index Terms*— wavelet transforms, feature extraction, neural networks, clustering methods, curve fitting**

I. Introduction

In many applications in the areas of climatology, hydrology, water resources, and agriculture, high spatial and temporal rainfall estimates are needed. Despite the fact that ground radars and rain gauges can provide very accurate estimates with high temporal and spatial resolution, they cannot cover the areas like oceans, mountain ranges and tropical rain forest which make up a large area of the globe. For these areas, satellite precipitation estimation can be a solution.

Different sensors such as active and passive radar, infrared, and lightning have been applied in precipitation estimation algorithms. Even though active and passive microwave sensors on satellites can provide physical information about clouds, their temporal resolution is not appropriate for high temporal applications. Moreover, infrared sensors can provide high temporal observation but their information is not physically connected to precipitation [1]. Because the infrared sensors can obtain only the cloud-top temperature observation, the algorithms solely based on infrared data cannot provide accurate estimations of precipitation, especially on a small scale [2]. Studies show that using infrared data with radar calibration can provide more accurate estimation [3-6].

Satellite precipitation algorithms can also be categorized into three groups: cloud-pixel-based, cloud-local-texture-based, and cloud-patch–based algorithms base on the level of IR information. In cloud-pixel-based algorithms, a rain rate (constant or variable) is allocated to every pixel of the cloud. In the cloud-local-texture-based methods, a rain rate is calculated and assigned to a pixel by considering a range of the neighborhood pixel coverage. Lastly, cloud-patch-based techniques use cloud coverage based on a specified temperature threshold [7].

II. Data

The study region covers an area of the United States extending between 30N to 38N and -95E to - 85E. The first 10 days of January 2008 are used for training and five days of that month from 15th to 20th are used for testing. The IR brightness temperature observations are from the GOES-12 satellite. The National Weather Service Next Generation Weather Radar (NEXRAD) Stage IV precipitation product produced by the National Centers for Environmental Prediction (NCEP) is used for training and validation. Also the PERSIANN-CCS precipitation estimates (obtained from the PERSIANN group) are used for comparing the results. The IR data from GOES-12 (Channel 4) has 30-minute interval images that cover the entire area of study. It also has a spatial resolution of 4 km × 4 km. The spatial resolution for NEXRAD Stage IV is 4 km × 4 km and the data are available as 1, 6, and 24 hourly accumulated precipitation values over United States.

III. Methodology

The satellite rainfall estimation algorithm studied in this research is based on a cloud-patch classification. First, the clouds are segmented into patches. Next, the features of each patch are extracted. Then, the patches are classified into groups. And finally, for each cluster, a proper relationship of temperature - rain-rate is assigned. The following section gives detail.

A. Cloud Segmentation

There are several methods for cloud-patch segmentation. We use seeded region growing segmentation [8]. In this method, first the minimum temperature of the clouds is determined. The minimum temperature can be more than one pixel. These minimum temperature pixels are seeds. By increasing step by step one degree temperature threshold from the minimum, the corresponding cloud pixels (which have the temperature equal or less the increased threshold) are

determined. If these pixels are neighbors of a seed they are considered belonging to the area of the seed. If not, that pixel is considered as a new seed. The threshold of the temperature increase until the temperature reaches 255K. Afterward a morphological operation is performed to remove/merge tiny region.

Fig.1.a (left), shows a cloud-top brightness temperature from GOES-12. The corresponding cloud patch segmentation using region growing is depicted in Fig. 1.b.

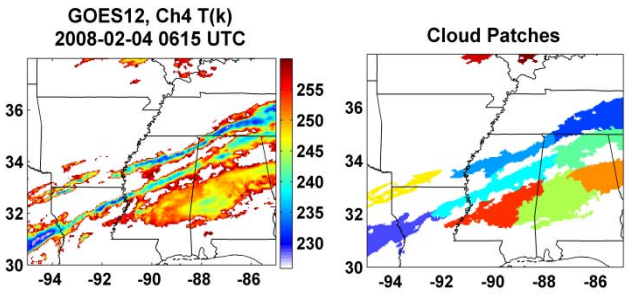

Fig.1. (a-Left) Cloud-top brightness temperature, (b-right) corresponding patches and segmented clouds

B. Feature Extraction

Similar to the PERSIANN-CCS algorithm, the coldness (statistic), geometry, and texture features of cloud patches can be used to differentiate cloud types. The feature extraction is applied to the patches for the thresholds of 220, 235, and 255K. The convective clouds usually have high local temperature variations and tight temperature gradients. Conversely, stratiform clouds have low temperature variations with gradual temperature gradients [7]. Along with PRESIANN-CSS features, wavelet features of different wavelet families and sliding window sizes are extracted from each patch. The wavelet families examined are Haar, Daubechies, Symlets, Coiflets, BiorSplines, ReverseBior, and DMeyer. Furthermore, sliding window sizes of 3 , 5, 7, and 9 are also investigated to find the optimal wavelet features. In the following section the wavelet features are mathematically described.

C. Wavelet Features

Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small waves called wavelets, of varying frequency and limited duration. This time-limited basis function helps to know what frequency and when it is used in expansion.

Equation (1) shows a 1D-continous-time function which is expanded by wavelet basis function in time and frequency [9]:

$$f(t) = \sum_k \sum_j a_{j,k}\psi_{j,k}(t) \qquad (1)$$

Where $f(t) \in L_2(R)$, $k \in Z$, and $j \in Z$. k is time and j is frequency (or translating variables and scale, respectively). The basis function is also defined as follows:

$$\psi_{j,k}(t) = 2^{\frac{j}{2}}\psi(2^j t - k) \qquad (2)$$

where $\psi(t)$ is a mother wavelet and $a_{j,k}$ is a wavelet coefficient. We can also expand (1) into the following formula

$$f(t) = \sum_k c_{j_0}(k)\Phi_{j_0,k}(t) + \sum_{j=j_0}\sum_k d_j(k)\psi_{j,k}(t) \qquad (3)$$

Where the first term of $f(t)$ is coarse (or approximation) resolution and the second term is detail (or wavelet) resolution of the signal. $\Phi(t)$ is a scaling function and is similar to wavelet function (2). $c_{j_0}(k)$ and $d_j(k)$ are approximation (or scaling coefficients) and detail (or wavelet) coefficients in scale j , respectively. They can be calculated by the following equations:

$$\begin{aligned} c_{j_0}(k) &= < f(t), \Phi_{j_0,k}(t) > = \int f(t)\,\Phi_{j_0,k}(t)\,dt \\ d_j(k) &= < f(t), \psi_{j,k}(t) > = \int f(t)\,\psi_{j,k}(t)\,dt \end{aligned} \qquad (4)$$

Note that j_0 is an arbitrary starting scale.

The above formulas are for one-dimension signals. To obtain wavelet coefficient of two-dimension signals like an image, we have scaling and wavelet function as follows [9]:

$$\begin{aligned} \Phi(x,y) &= \Phi(x)\Phi(y) \\ \psi^H(x,y) &= \psi(x)\Phi(y) \\ \psi^V(x,y) &= \Phi(x)\psi(y) \\ \psi^D(x,y) &= \psi(x)\psi(y) \end{aligned} \qquad (5)$$

One two-dimensional scaling function $\Phi(x,y)$ and three two-dimensional wavelet functions $\psi^H(x,y)$, $\psi^V(x,y)$, $\psi^D(x,y)$ are required for expanding a two dimensional signal by wavelet transform. The two-dimensional scaling and wavelet function have the same one-dimensional definition, so we have:

$$\begin{aligned} \Phi_{j,k,l}(x,y) &= 2^{\frac{j}{2}}\Phi(2^j x - k, 2^j y - l) \\ &= 2^{\frac{j}{2}}\Phi(2^j x - k)\Phi(2^j y - l) \end{aligned} \qquad (6)$$

The same formula we can use for wavelet functions.

Similar to (3) we can obtain the following two-dimension function expanded using scale and wavelet function:

$$\begin{aligned} f(x,y) = &\sum_{k,l} c_{j_0}(k,l)\Phi_{j_0,k,l}(x,y) \\ &+ \sum_{s=H,V,D}\sum_{j=j_0}\sum_{k,l} d_j^s(k,l)\psi_{j,k,l}^s(x,y) \end{aligned} \qquad (7)$$

Where $c_{j_0}(k,l)$ is scaling (or approximation) coefficient and $d_j^s(k,l)$ is Horizontal, Vertical, Diagonal wavelet coefficients (s = H, V, D respectively).

Fig. 2, shows an example of using the wavelet transform to obtain approximation and wavelet coefficients at scales (level) 1and 2 from Daubechies (db1) wavelet with sliding window

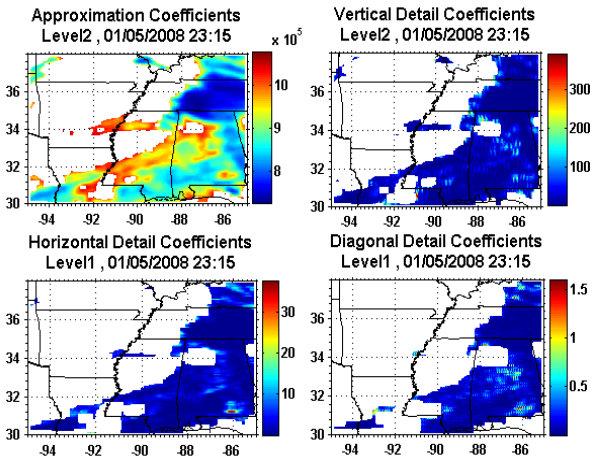

Fig. 2. Approximation and vertical detail Wavelet coefficients for level two (top left and right) and horizontal and diagonal detail wavelet coefficients for level one (bottom left and right)

size 7. The wavelet features are mean and standard deviation of the detail coefficients' energies in the horizontal, vertical, and diagonal orientation for two levels.

D. Classification and Clustering

An unsupervised neural network classification, self-organizing map (SOM) neural network [10], is used in this study for cloud classification. SOM projects patterns of high dimensional space to lower dimensional space. The projection enables the input patterns of many variables to be classified into a number of clusters and to be arranged in a two dimensional coordinate. In this technique, the distances between the input pattern (features) and cluster weights are computed and the cluster with the minimum distance is selected. Through a recursive equation, the selected cluster weight and its neighbor's weight are adjusted. The self-organizing map algorithm can be explained as a series of 9 steps:

Step0: Initializing the weights matrix, setting the topological neighborhood, setting the learning rate.
Step1: Setting the stop condition; if the condition is viable, proceed with the following steps:
Step2: For each input vector, follow steps 3-5
Step3: For each input, compute the distance between the input and every weight that Corresponds output neurons
Step4: Find the minimum distance and the index having the minimum distance
Step5: Compute the new weights or all neurons within a specified neighborhood of the minimum distance neuron

$$w_{ij}(new) = w_{ij}(old) + \alpha[x_i - w_{ij}(old)] \qquad (8)$$

Step6: Update the learning rate
Step7: Repeat for some epochs
Step8: Reduce the radius topological neighborhood after the epochs.
Step9: Check the test stopping condition.

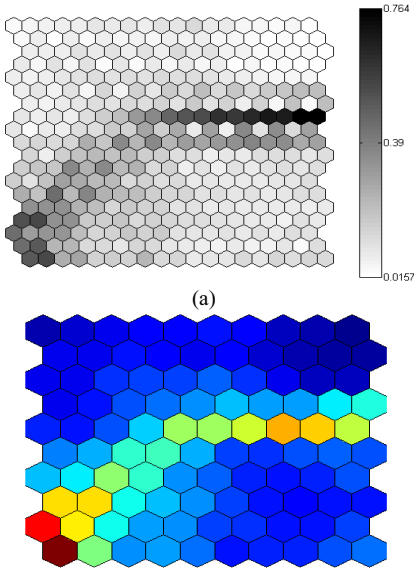

Fig. 3. Distance between clusters and neighbors (a) distance shows by hexagonal (b) average distance shows by color for each cluster

In this study, a 10×10 unit map with hexagonal structure is used as clusters. By using the above SOM algorithm, the weights for each cluster are obtained. Fig. 3.a, shows the cluster distance between each unit and its neighbors. The more dark value, the more distance between two clusters. Each hexagonal between two cluster shows the distance between two clusters. Also Fig. 3.b, shows the clusters and the average distance between them and their neighbors.

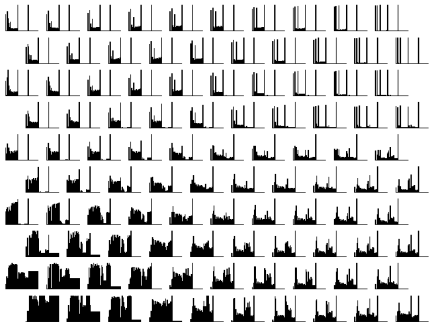

Fig. 4. Proportion of each feature to each cluster

In addition, Fig. 4, shows the relative proportion of each feature to each cluster (each tiny bar indicates a feature). As it is seen, almost all features are involved in the clusters at bottom left. One of the advantages of SOM is to monitor and observe the values of features for each clusters. Fig. 5, shows the values of some features for each cluster. In these figures, blue is the lowest and dark red is the highest value. For example Tmin (a minimum temperature of a patch cloud) for bottom clusters are lowest. By looking at the values of Tmin, Tmean , STD and area, we can notice the bottom clusters are representing the convective clouds and the top clusters are warm clouds. The wavelet features also are high value at bottom left.

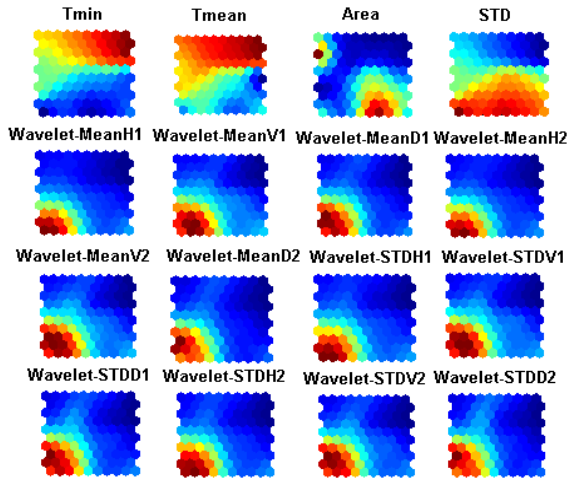

Fig. 5. Feature values for clusters

E. Assigning the Cluster Rain Rate

In this section, a T-R (Temperature–Rain Rate) curve is assigned to each cluster. In order to obtain this curve, first T-R pixel pairs (obtained from GOES-12 observation and Nexrad data) are redistributed by using the Probability Matching Method (PMM) [11]. Then, a non-linear least squares method is used to fit the T-R data to a polynomial function with degree 10.

IV. Results And Validation

Fig. 6, shows a block diagram of an implementation of this algorithm. First, the data from GOES-12 is calibrated and converted to an image as a pre-processing step of the implementation.

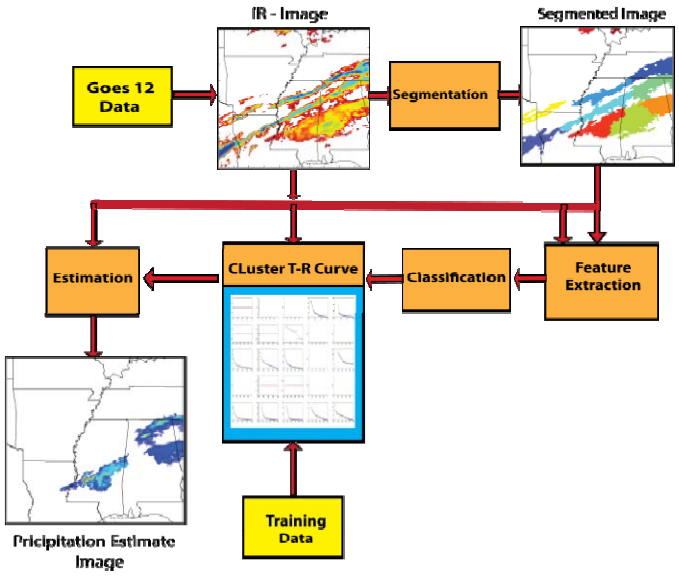

Fig. 6. Model Structure for an implementation of the algorithm

Then, using the region growing method the clouds are segmented and divided into patches. The texture, coldness, and geometric features are extracted from each patch in the next step. In step 3, a neural network classifier is used to cluster the input patterns. Finally, if we are in training mode, the T-R curve is assigned to each cluster; otherwise, a T-R curve of selected cluster is used to estimate rainfall of the entry patch.

A. Evaluation and comparison of results

Our goal in this paper is to find the optimal wavelet features in terms of wavelet family and the sliding window size.
The wavelet families considered in this research are Haar, Daubechies (‘db4’, ‘db5’), Symlets (‘sym2’), Coiflets (‘coif1’), BiorSplines (‘bior1.1’), ReverseBior (‘rbio1.1’), and DMeyer (‘dmey’). Fig. 7, shows an example of hourly rainfall estimate using these wavelet families. Note that the family wavelets Coiflets, ReverseBior, Daubechies (db4, db5) have the same estimate result; So we just show Daubechies (‘db4’) as a representative of them. In addition, the PERSIANN_CCS rainfall estimate and NEXRAD IV data are depicted in the bottom of the image as comparing and reference data, respectively. As it is seen the Symlets, Daubechies (db4), and BiorSplines wavelets are more similar together in terms of rainfall estimate than that of

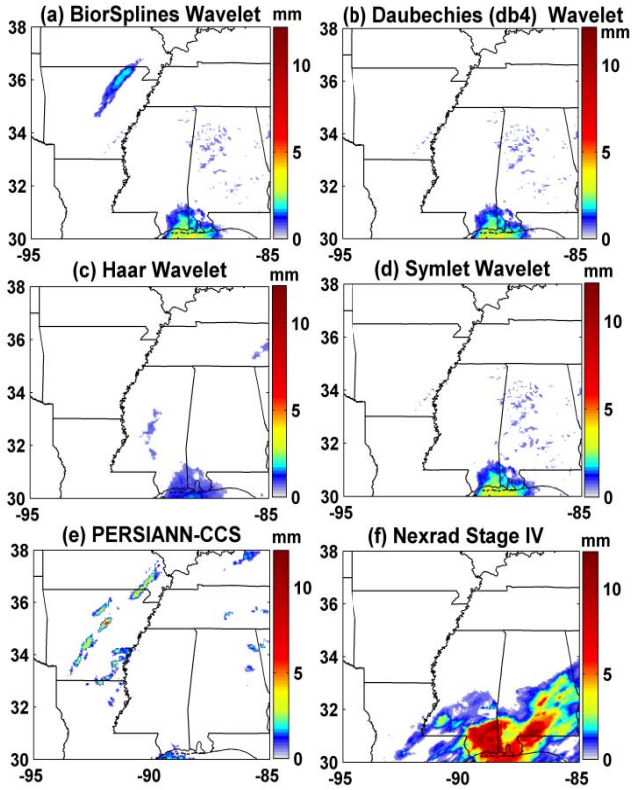

Fig. 7. Estimated hourly rainfall estimates for ending at 1500 UTC on January 19, 2008: (a) BiorSplines, (b) Daubechies (db4), (c) Haar, (d) Symlets, (e) PERSIANN-CCS, (f) Nexrad Stage IV.

Haar wavelet. The results also show that using wavelet features has better performance than the PERSIANN_CCS estimate if compared to NEXRAD IV.

B. Validation

Several evaluation criteria are selected to validate the results. The quantitative accuracy of the estimates is evaluated using the root mean squared error (RMSE). The performance of rain/no-rain detection is evaluated by the probability of detection (POD), the false-alarm ratio (FAR), and the Heidke skill score (HSS). All the estimate results are validated against NEXRAD Stage IV. Fig. 8, illustrate the HSS, POD, FAR, and RMSE performance measures for different wavelet families. In the case of rain /no-rain detection, we see Haar and symlet mother wavelet have better performance. (for simplicity, Haar, Daubechies, Symlets, BiorSplines, and PERSIANN-CCS are refered to as 'haar', 'db4', 'sym', bio'', 'CCS'). In terms of the quantitative evaluation, the Haar and Symlet mother wavelets work better than others as well. Note that for every wavelet family, the maximum composition level is considered.

Likewise, we see in these metrics that all wavelet estimates have better performance than PERSIANN-CCS.

Fig. 9, shows the performance of different sliding window sizes used for wavelet feature extraction. The window sizes applied are 3, 5, 7, and 9. As it is seen, window size 5×5 has better HSS than other sizes. That is, this window size works better in case of rain/no-rain detection. In addition, in terms of RMSE, we see less RMSE when window size 5×5 and 9×9 are used. In the other words, sliding window size 5×5 performs better than other sizes in terms of quantity estimation accuracy.

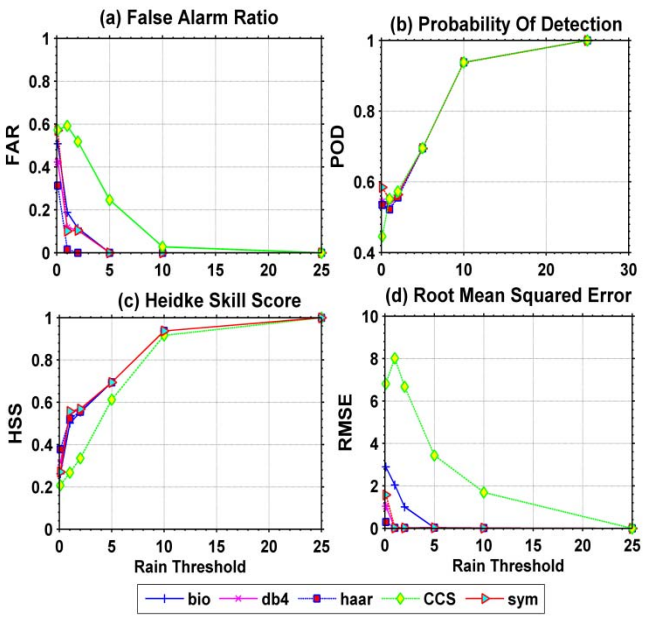

Fig. 8. Verification results for different wavelet families: (a) False Alarm Ratio; (b) Probability Of Detection; (c) Heidke Skill Score; (d) Root Mean Squared Error

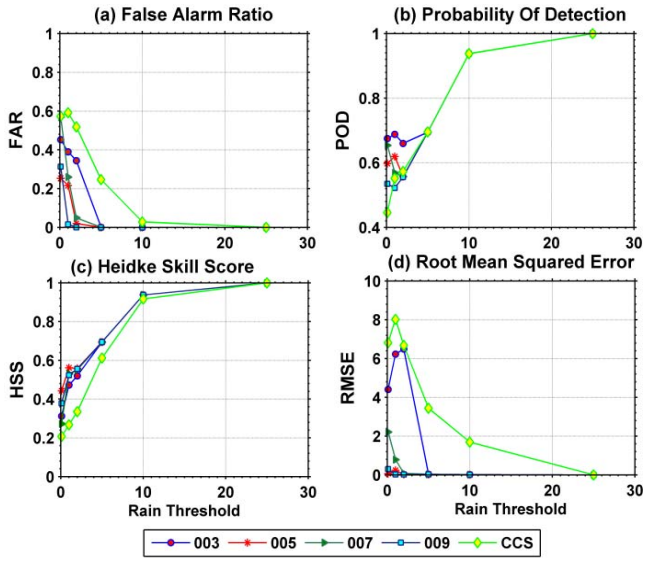

Fig. 9. Verification results for different sliding window sizes : (a) False Alarm Ratio; (b) Probability Of Detection; (c) Heidke Skill Score; (d) Root Mean Squared Error

Generally speaking, looking at the rain/no-rain metrics and quantity evaluation, the sliding window size 5×5 has better result than other sliding window sizes.

V. SUMMARY

In this study, we have examined different wavelet families, as well as different sliding window sizes, to find the optimal wavelet features for cloud classification and rainfall estimation in a satellite precipitation estimation algorithm based on wavelet features. To implement this algorithm requires four steps: cloud-patch segmentation, feature extraction, cloud patches clustering, and assigning Temperature-Rainfall curve to each cluster. The result shows that Haar and Symlet family wavelets have better performance than other family wavelets. Furthermore, the sliding window size 5×5 presents better result than other sizes in terms of HSS and RMSE.

REFERENCES

[1]. R.R Ferraro, and G. Marks, "The development of SSM/I rainrate algorithms using ground-based radar measurements". *J. Atmos. Oceanic Technol*., 12, 755–770, 1995

[2] P.A. Arkin and B. N. Meisner, "The relationship between large-scale convective rainfall and cold cloud over the Western-Hemisphere during 1982-84," *Monthly Weather Review,* 115, 51-74, 1987

[3] Huffman, G. J., R. F. Adler, D. T. Bolvin, G. J. Gu, E. J. Nelkin, K. P. Bowman, Y. Hong, E. F.Stocker, and D. B. Wolff, " The TRMM multisatellite precipitation analysis (TMPA): Quasi-global, multiyear, combined-sensor precipitation estimates at fine scales", *Journal of Hydrometeorology*, 8, 38-55. 2007

[4]. Tim. Bellerby, K-L Hsu, S. Sorooshian, "LMODEL A Satellite Precipitation Algorithm Using Cloud Development Modeling and Model Updating: Part 1: Model Development and Calibration", *J. of Hydrometeor*,10, 1081–1095, 2009

[5] K. L., Hsu, X. G. Gao, S. Sorooshian, and H. V. Gupta, "Precipitation estimation from remotely sensed information using artificial neural networks". *Journal of Applied Meteorology*, 36, 1176-1190, 1997

[6] Joyce, R. J., J. E. Janowiak, P. A. Arkin, and P. Xie, " CMORPH: A method that produces global precipitation estimates from passive microwave and infrared data at high spatial and temporal resolution", *Journal of Hydrometeorology*, 5, 487-503, 2004

[7] Y. Hong, K. L. Hsu, S. Sorooshian, and X. G. Gao, "Precipitation Estimation from Remotely Sensed Imagery using an Artificial Neural Network Cloud Classification System," *Journal of Applied Meteorology,* 43, 1834-1852, 2004

[8] Gonzalez, R. C., R. E. Woods, *Digital Image Processing*, Prentice Hall (3th edition), August 2007

[9] Burrus C.S., R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms: A Primer*, Prentice Hall, August 24, 1997

[10] Kohonen, T., "Self-organized formation of topologically correct features maps," *Biol. Cybernetics,* 43, 59–69, 1982

[11] D. Atlas, D. Rosenfeld, and D.B. Wolff, "Climatologically tuned reflectivity-rain rate relations and links to area-time integrals," *J. Appl. Meteor*., 29, 1120-1135, 1990