International Journal of
Advanced Science and Convergence

ISSN : 2619-8150
Volume 2, Number 3, September 2020
https://doi.org/10.22662/IJASC.2020.2.3.021

Raining State Study using Gaussian Mixture Model

Jeongwon Lee[1] and Choong Ho Lee[2]

[1]Student, Department of Information Communication and Engineering, Hanbat National University, Daejoen, 34158, Republic of Korea

[2] Professor, Department of Information Communication and Engineering, Hanbat National University, Daejoen, 34158, Republic of Korea

Abstract

Background/Objectives: In this study, we intend to extract a characteristic study of the raining situation based on the GMM (Gaussian mix model) within the raining image and perform the raining phenomenon in real time. **Methods/Statistical analysis**: Since the Gaussian Mixture Model is a good model for expressing natural phenomena, the Gaussian Mixture Model was used in many fields. In this study, an OpenCV library provided by a Python program was used to analyze rain video using CV2 library. **Findings**: A test video image object was created, and the frame size was set to create a gray scale image and an image for accumulating color frames. A video frame was acquired, and an average image was calculated using the accumulated images for color image frames and gray scale images, and displayed in a window. An error rate occurs during the initial image analysis, but the accuracy increases over time. By analyzing the raining image, the motion of the rain was detected using the findObjectAndDraw() function to remove the noise. In the image of the detected rain, it was marked with a red bounding rectangle to confirm the rain. **Improvements/Applications**: In this study, a Gaussian mixture model was used, but the K-NN model was applied to make a more accurate analysis through comparison between various models.

Index Terms

GMM, Image Analysis, Motion Analysis, Computer Vision, Gaussian Model

Corresponding author : Choong Ho Lee
chlee@hanbat.ac.kr

• Manuscript received August 3, 2020.
• Revised September 2, 2020 ; Accepted September 21, 2020.
• Date of publication September 30, 2020.

I. Introduction

With the development of the modern information society, the desire to monitor the environment is increasing. Various environmental monitoring technologies are being developed to meet these needs.

To prove the difference between pixels, modeling of the Gaussian mixture model (GMM) has been proven to be the most effective method for monitoring remote locations [1]. The Gaussian mixture model (GMM) has the advantage of having a simple structure and being able to effectively model the characteristics in an image even with a small amount of image [2-3].

In this study, we tried to study the raining phenomenon in real time by extracting characteristic parameters for the raining situation based on GMM (Gaussian mixture model) from the raining image.

II. Theoretical Background

A. GMM(Gaussian mixture model)

The mixed model is a probabilistic model for indicating the existence of subgroups within the whole group in statistics. More formally, it corresponds to a mixed distribution representing the probability distribution of the entire population. Mixed distribution problems are related to obtaining the characteristics of the whole group from the subgroups, whereas mixed models are used to make statistical inferences about the characteristics of the subgroups given the observed group [4-5].

If you define the combined distribution of the observed and latent variables of the mixed model, the distribution of the observed variables can be marginalized for all latent variables. The complex distribution of the observed variables can be expressed more simply using latent variables, and a mixed model is used to estimate the complex distribution by assuming these latent variables. This mixed model is used to cluster the data [6]. Expectation-Maximization algorithm is used for Maximum-Likelihood-Estimation of a model including latent variables.

The Gaussian mixture model (GMM) is widely used in data mining, pattern recognition, machine learning, and statistical analysis.

B. Mixed model background

1. Supervised Learning

Supervised Learning consists of a pair of variables {x, ω} consisting of a feature vector (x) and a class to which the observed value belongs (ω). Learning is called Supervised Learning, a teacher, a supervisor, and a feature vector and an accurate answer.

2. Unsupervised Learning

Class label (ω) is not given and feature vector x=(x1, x2,... When a data set consisting of only, xN} is given, it is called non-teacher, unsupervised, and unsupervised learning because the correct answer is not provided.

Two approaches of Unsupervised Learning are Parametric mixed model method using mixed model and Non-Parametric method. “Fig.1” is a graph of Gaussian Mixture Model, a representative method of Parametric method among the approaches of unsupervised learning [7-9].

1. Parametric mixed model method

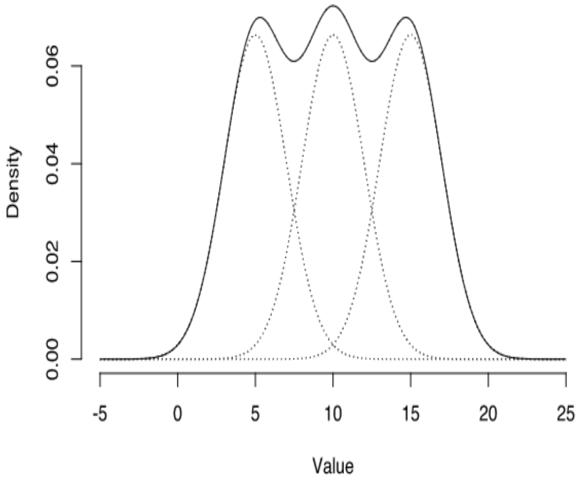


Fig. 1. Gaussian Mixture Model graph, a representative method of Parametric method among unsupervised learning approaches

The parametric mixing model method is modeled by mixing several parametric probability density functions, Gaussian, to model a given probability density function. Parametric hybrid models aim to find model parameters.

2. Non-Parametric method

The non-parametric method is called clustering (clustering) by dividing the data into a fixed number of clusters without making any assumptions about the given data.

III. Research Analysis

A. Analysis Method

Since the Gaussian Mixture Model is a good model for expressing natural phenomena, the Gaussian Mixture Model was used in many fields.

It can be expressed as the following equation as the sum of the K Gaussian distributions of functions to be optimized, and can be expressed as several Gaussian distributions as shown in "Fig. 2".

$$p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k)$$

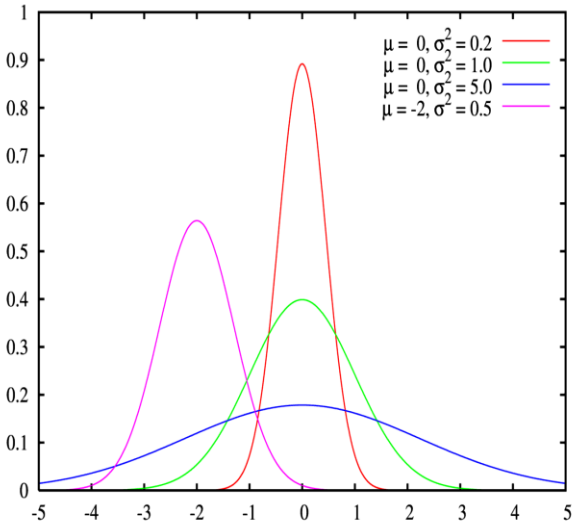


Fig. 2. Multiple Gaussian distributions

The image used in this study is an image of rain on a fixed background, and is shown in "Fig. 3".

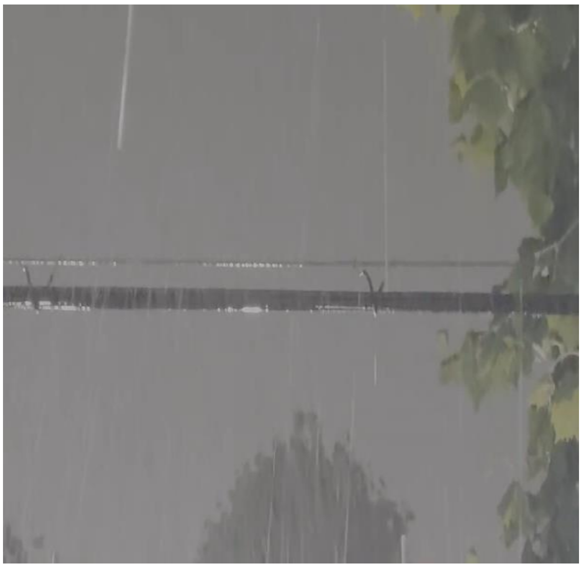


Fig. 3. Raining video used in this study

B. Analysis Tools

In this study, Python and OpenCV programs, which are effective for image analysis, were used as shown in "Fig. 4" below to analyze rain conditions.

Fig. 4. Python and OpenCV Programming

Python programs are high-level programming languages, platform-independent, interpreted, object-oriented, and dynamic typing interactive languages. OpenCV (Open source Computer Vision) is a library created for the purpose of processing real-time computer vision and provides a wide variety of functions such as image, image processing, object detection, motion detection, and so on.

C. Analysis Library

In this paper, an OpenCV library provided by a Python program was used to analyze the rain video using the CV2 library.

It is very important for the CV2 library to divide the foreground object from the background in order to divide the image from the video. Background-Subtraction is the simplest video image segmentation method and calculates the pixel difference between the background image and the current input frame image. It is important to determine the pixel position above the threshold value as a pixel with change and calculate it stably [10].

In this study, Accumulate(), AccumulateSquare(), AccumulateProduct(), AccumulateWeighted() accumulated image functions provided by CV2 library are used as shown in "Table.1".

The CV2 library can generate a background image using a cumulative image function, and when calculating a background image using a moving average, there are many cases where a moving object is in the first frame of the video. Since the moving object may remain in the background, the moving average is calculated after calculating the background image with the Accumulate() function for a certain period of time.

Set the mask image maks(x, y) = 0 of the pixel whose absolute value of the difference image with the background image is greater than the threshold value, and if the absolute value of the difference image is less than the threshold value, the mask image maks(x, y) of the pixel) = 1, and only the pixels that gradually change smoothly are updated with the moving average.

Table 1. OpenCV Cumulative Image Function Provided by Python Programming

Function	Function details
Accumulate(src, dst[, mask]) -> dst	Image accumulation
AccumulateSquare(src, dst[, mask]) -> dst	image accumulation of squared images
AccumulateProduct(src1, src2, dst[, mask]) -> dst	Image accumulation of multiply images
AccumulateWeighted(src, dst, alpha[, mask]) -> dst	Moving average image

The input image src is a real image of 8 bits or 32 bits of 1 or 3 channels. The accumulated image dst has the same number of channels as src and is a real image of 32 bits or 64 bits. In the case of 3-channel video, it is added independently for each channel.

The Accumulate() cumulative image function accumulates the pixels of the image, and AccumulateSquare() accumulates the pixels of the image by squared them. Average and variance images are calculated with Accumulate() and AccumulateSquare(). AccumulateWeighted() calculates the Running Average with dst=(1-alpha) × dst + alpha × src.

D. Analysis Result

Image pixels were modeled using the statistical method of the Gaussian mixture model, and the foreground pixels were segmented from the background. For the modeling of each pixel, the OpenCV library provided by Python, BackgroundSubtractor and BackgroundSubtractorMOG2 classes were used, and the BackgroundSubtractorMOG2 class modeled each pixel of the image with a Gaussian mixture model.

The findObjectAndDraw() function was used to remove noise and detect contours in binary images by morphological operation, and to detect objects larger than the desired size, display the red bounding rectangle in the image res copied from the src, and return it.

By applying Gaussian blurring to each frame of the input image, an object was detected in each binary projection that divided the moving object in the foreground from the background, and an image with a red bounding rectangle was created and displayed on the window.

In the first part of the video, many false detections occur. “Fig. 5” and “Fig. 6” are images divided into binary with Gaussian mixture model, and are the result of raining by detecting difference images.

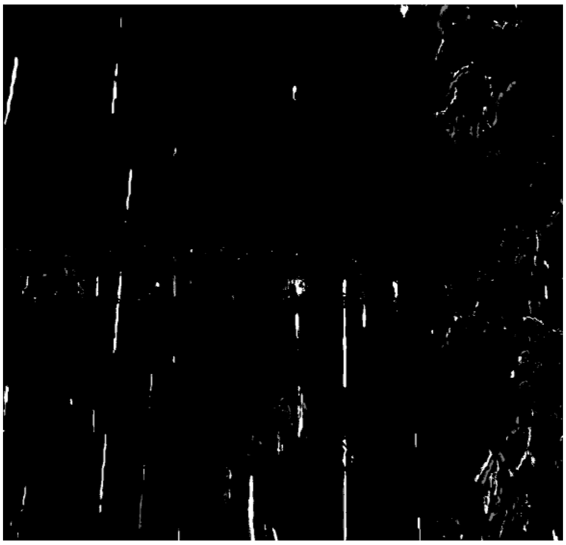


Fig. 5. Gray video of rain

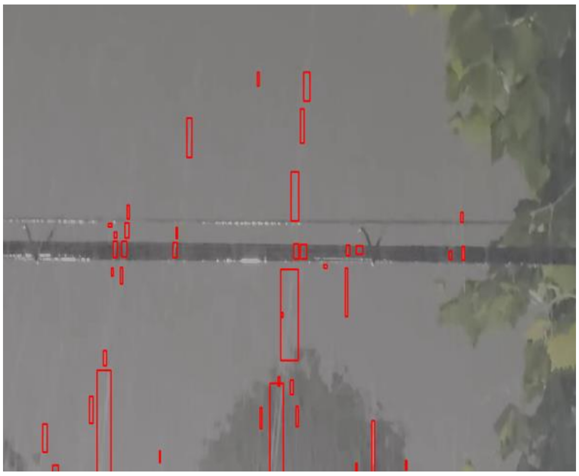
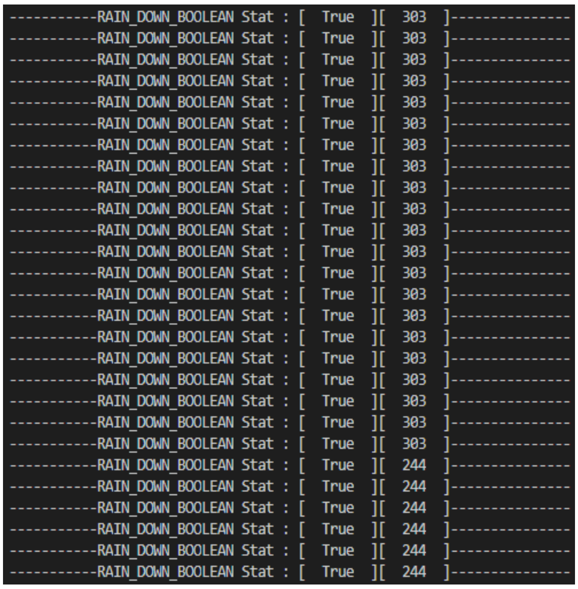


Fig. 6. Color video of rain

A test video image object was created, and the frame size was set to create a gray scale image and an image for accumulating color frames. A video frame was acquired, and an average image was calculated using the accumulated images for color image frames and gray scale images, and displayed in a window. An error rate occurs during the initial image analysis, but the accuracy increases over time.

By analyzing the raining image, the motion of the rain was detected using the findObjectAndDraw() function to remove the noise. In the image of the detected rain, it was marked with a red bounding rectangle to confirm the rain. ″Fig. 7” is a result of monitoring the detection status of rainy images, and indicates True in case of rain, and the number of detections is indicated by numbers.



```
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  303  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  244  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  244  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  244  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  244  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  244  ]--------------
----------RAIN_DOWN_BOOLEAN Stat : [  True  ][  244  ]--------------
```

Fig. 7. Detection status of raining video

IV. Conclusion and Suggestions

An image captured from a remote location was analyzed using the difference between pixels using the GMM(Gaussian mixture model) modeling technique. The Gaussian mixture model has a simple structure and can effectively model characteristics in images even with a small amount of images.

In this study, we succeeded in grasping the raining condition in real time by extracting characteristic parameters for the raining situation based on GMM from the raining image. However, as the most effective method for analysis, not only the Gaussian mixture model, but also a method of modeling pixels using a statistical method using K-NN (K-nearest neigbours), and a method of segmenting the foreground pixels from the background.

In this study, a Gaussian mixture model was used to confirm the raining condition, but the K-NN model was applied to make a more accurate analysis through comparison between various models.

References

[1] S. H. Chin, & J. H. Lee, & C. H. Kang. (2000). A Study on Adaptive Model Updating and a Priori Threshold Decision for Speaker Verification System. *The journal of the acoustical society of Korea*, 19(5), 20-26.

[2] D. A. Reynolds, & R. C. Rose. (1995). Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. *Speech and Audio Processing*, 3, 72-83.

[3] W. W. Hung, & H. C. Wang. (2001). On the Use of Weighted Filter Bank Analysis for the Deviation of Roust MFCCs. *IEEE signal processing Letters*, 8(3).

[4] A. M. Mclvor. (2000). Background subtraction techniques. *Proc. of Image and Vision Compouting*, 1(3), 155-163.

[5] Z. Mayo, & J. R. Tapamo. (2009). Background Subtraction Survey for Highway Surveilance. *in Proc. Annu. Symp. PRASA*, 77-82.

[6] S. S. Cheung, & K. Chandrika. (2004). Robust techniques for background subtraction in urban traffic video. *proc. of SPIE*, 5308(1), 881-892.

[7] C. Stauffer, & W. Grimson. (1999). Adaptive background mixture models for real-time tracking. *IEEE Computer Society Conference on*, 2, 246-252.

[8] C. Stauffer, & W. Grimson. (2000). Learning Patterns of Activity Using Real-time Tracking. *IEEE Trans. on PAMI*, 22(8), 747-757.

[9] T. Bouwmans, & F. El Baf, & B. Vachon. (2008). Background modeling using mixture of gaussians for foreground detection-a survey. *Recent Patents on Computer Science*, 3(1), 219-237.

[10] P.W. Power, & J.A. Schoonees. (2002). Understanding background mixture models for foreground segmentation. *Proc. of image and vision computing*, 2002, 266-271.