Logo

Description automatically generated

Genetic Algorithm to Optimize the Constant Area Coding Loseless Compression

CSE489: Selected Topics in Data Science

**Supervised By:**

Prof. Dr. Alaa Hamdy

**Submitted By:**

Youssef George Fouad Saad 19P9824

**Semester:**

Spring 2024

# Executive Summary

In the dynamic field of computer vision, accurate and real-time foreground detection remains critical for applications ranging from surveillance to interactive systems. This project aimed to implement and analyze an adaptive Gaussian Mixture Model (GMM) for foreground masking, as described in the seminal paper by Stauffer and Grimson. The goal was to develop a robust method that dynamically updates the distributions of Gaussian models for each pixel, thereby differentiating between background and foreground movements in real-time RGB video streams.

The adaptive GMM was implemented in Python, where each pixel in the video frame was modeled as a mixture of Gaussian distributions. These distributions were continuously updated based on the pixel values observed over time, which allowed for real-time adjustments to changes in the scene's lighting, object movement, and other environmental variables. This approach was tested against traditional methods such as frame averaging, where the efficacy of handling scenes with varying lighting and moving objects was compared.

The implementation of an adaptive GMM for foreground masking has demonstrated significant advantages in terms of flexibility, accuracy, and real-time processing capabilities. The system's ability to adapt to a wide range of environmental conditions without the need for frequent re-calibrations makes it a robust solution for real-time video analysis tasks. Future work will explore optimizations in computational efficiency and model accuracy, potentially extending the system's applicability to other types of sensory data beyond RGB images.

Table of Content

Executive Summary ii

1 Introduction 1

1.1 Motivation 1

1.2 Objective 1

1.3 Scope 1

1.4 Report Structure 2

2 Methodology 3

2.1 Gaussian Mixture Model 3

2.2 Adaptive Learning 3

2.3 Real-Time Implementation 4

2.4 Comparison with Frame Averaging 4

3 Experimental Setup 5

3.1 Dataset 5

3.2 Preprocessing 6

3.3 Implementation Details 6

3.4 Model Parameters 7

4 Results and Discussion 8

5 Comparison with Other Techniques 9

# Introduction

Foreground detection in video streams is a fundamental task in the field of computer vision, with applications spanning security surveillance, traffic monitoring, interactive media, and more. The challenge lies in accurately distinguishing between static background and moving foreground objects in real-time, under varying lighting conditions and dynamic scene changes. Traditional methods often fall short in complex environments, leading to the need for more adaptive and robust approaches.

## Motivation

This project was motivated by the limitations of conventional foreground detection methods, which typically employ static background subtraction or simple frame averaging techniques. These methods often require manual recalibration and struggle with lighting variations, slow-moving objects, and dynamic backgrounds. The aim was to implement a more flexible and autonomous system that could adapt in real-time to changes within the video feed, minimizing human intervention and improving detection accuracy based on the paper by Stauffer and Grimson [1].

## Objective

The primary objective of this project was to implement and evaluate an adaptive Gaussian Mixture Model (GMM) for foreground masking based on the method described by Stauffer and Grimson. This model represents each pixel as a mixture of Gaussian distributions, which are continuously updated. The adaptability of the GMM allows it to handle real-world complexities such as sudden lighting changes more effectively than traditional methods.

## Scope

The project involved:

- Developing a Python-based implementation of the adaptive GMM for real-time video processing.

- Testing and optimizing the system across various hyperparameters to balance responsiveness and computational efficiency.

- Comparing the performance of the adaptive GMM against traditional foreground detection methods like frame averaging, particularly in scenarios with variable lighting and multiple moving objects.

## Report Structure

The remainder of this report details the methodology employed in implementing the adaptive GMM, discusses the experimental setup and results from testing the system under different conditions, and compares these results with those obtained from conventional methods.

# Methodology

## Gaussian Mixture Model

The GMM is employed to represent the values of a particular pixel over time as a mixture of multiple Gaussian distributions. This approach accounts for different surface reflections and lighting variations captured in video sequences.

* **Model Initialization**: Each pixel starts with `K\_numOfGauss` Gaussian distributions. Parameters initialized are:
* **Means ():** Set to an intermediate gray value (e.g., 122 for all color channels).
* **Variances ():** Set to a starting variance (e.g., 36.0), assuming moderate initial variability.
* **Weights ():** Equally distributed among the distributions, starting with .

## Adaptive Learning

The GMM is adaptive; it updates its parameters based on each new frame processed, allowing it to handle real-world complexities such as changes in the background and the introduction of new objects.

**Parameter Update Rules:**

* **Weights ():** Decrease slightly for all distributions, with an increase for the matching distribution to reflect its higher likelihood given the new observation.
* **Means () and Variances ():** Updated using a learning factor alpha () which dictates the rate of adaptation to new data. This is based on the distance of the current pixel value from the means of the distributions, adjusted inversely by the covariances.

**Model Reordering and Background Thresholding:**

After updating the distributions, the model reorders them based on the ratio of weights to the square root of variances (). This heuristic helps in identifying the most probable distributions to be from the background, facilitating a faster and more accurate foreground detection.

**Background Determination:**

A cumulative probability threshold () is used to decide how many of the ordered distributions will be considered as part of the background. This is pivotal in adapting to scenes with varying foreground and background dynamics.

## Real-Time Implementation

The model is implemented in a Python environment using libraries such as NumPy for numerical operations and OpenCV for image processing tasks. The implementation details include:

* **Frame Processing:** Each frame is processed in sequence, where the model parameters are updated, and foreground regions are identified and marked.
* **Foreground Detection:** Based on the updated model, pixels that significantly deviate from the background model (as determined by the ) are classified as foreground.

## Comparison with Frame Averaging

Additionally, the methodology compares the adaptive GMM with a baseline frame averaging technique. This traditional method involves averaging the images over time to create a static background model, against which new frames are compared to detect changes.

A simple averaging method is used where the background is updated continuously by blending it with new frames, weighted by a small factor to adjust the rate at which the model adapts to new conditions.

# Experimental Setup

## Dataset

The dataset [2] used for evaluating the adaptive Gaussian Mixture Model (GMM) was selected based on its relevance to real-world applications involving foreground detection. The dataset consists of video sequences containing a mix of static backgrounds and moving foreground objects, providing a suitable environment to test the adaptability of the model to changing conditions.

* **Source:** The dataset was sourced from publicly available collections often used in computer vision research, specifically aimed at foreground-background separation.
* **Structure:** It contains sequences of video frames stored in (240\*360\*3) RGB images format, enabling frame-by-frame analysis.

A collage of a person standing in an office

Description automatically generated

Figure 1: Sample Images of the Dataset

A collage of images of a person

Description automatically generated

Figure 2: Sample Ground Truth Images

## Preprocessing

Each frame was resized to a standard dimension to ensure consistent input across the model and maintain computational efficiency.

* **Resizing:** Frames were resized to a resolution of 200\*300 pixels. This decision was driven by the need to balance between model performance and computational efficiency.
* **Color Channels:** Each frame was processed with three color channels (RGB) to utilize the full spectrum of pixel information.

## Implementation Details

The adaptive GMM was implemented using Python and libraries such as OpenCV and NumPy, enabling real-time processing and analysis.

* **Software:** The implementation was done in Python, utilizing Jupyter notebooks for interactive development and analysis.
* **Hardware:** The system was tested on a standard personal computer with sufficient computational resources to handle the video sequences in real-time.

## Model Parameters

The adaptive GMM was configured with specific parameter values to best adapt to the dataset:

* Number of Gaussians (K\_numOfGauss): Tried with 3 and 5 to balance the mixture complexity and computational efficiency.
* Background Threshold (BG\_thresh): Tried with 0.4 and 0.7 to identify the proportion of data considered as background.
* Learning Rate (alpha): Set to 0.01, allowing the model to adapt incrementally to changes in the video frames.

## Evaluation Metrics

To assess the performance of our adaptive Gaussian Mixture Model (GMM) for foreground masking, we employed several key evaluation metrics: Accuracy, Mean Absolute Error (MAE), and Intersection over Union (IoU).

### Accuracy

The proportion of correctly classified pixels (both foreground and background) to the total number of pixels. Accuracy provides an overall performance measure of the model. High accuracy indicates that the model correctly identifies most of the foreground and background pixels.

Where:

**TP (True Positives):** Number of foreground pixels correctly identified as foreground.

**TN (True Negatives):** Number of background pixels correctly identified as background.

**FP (False Positives):** Number of background pixels incorrectly identified as foreground.

**FN (False Negatives):** Number of foreground pixels incorrectly identified as background.

### Mean Absolute Error

MAE measures the average absolute difference between the predicted mask and the ground truth mask. It quantifies the prediction error of the model on a pixel-by-pixel basis. A lower MAE indicates that the model's predictions are closer to the actual values, thereby reflecting better performance in terms of pixel-wise accuracy.

Where:

**N:** the total number of pixels.

**Pi:** the predicted value of pixel i.

**Gi​:** the ground truth value of pixel i.

### Intersection Over Union

IoU, also known as the Jaccard index, measures the overlap between the predicted foreground mask and the ground truth foreground mask. It is a critical metric for evaluating the accuracy of the predicted segmentation masks. It considers both false positives and false negatives, providing a robust measure of the overlap between the predicted and actual foreground regions. A higher IoU indicates better performance, signifying that the model's predicted mask closely matches the ground truth.

# Results and Discussion

The adaptive Gaussian Mixture Model methodology offers a robust approach to foreground detection by accommodating changes in the video streams. This model's adaptability to new objects and changing conditions significantly surpasses traditional static methods, providing a more reliable solution for applications requiring precise and dynamic object detection.

Each of the above mentioned metrics provides a different perspective on the model's effectiveness in distinguishing between foreground and background in video frames.

Combining these metrics allows us to assess the model's strengths and weaknesses, guiding further improvements and optimizations. Our evaluation demonstrates that the adaptive GMM approach effectively handles dynamic scenes and varying lighting conditions, maintaining high performance across all three metrics.

# Comparison with Other Techniques

Another classical technique used for the same application of foreground object extraction is the averaging technique where the channel/pixel value is averaged as the frames goes on with a forgetting factor to allow for continuous update to the pixel average value.

In every time step, after updating the average value of the pixel, the difference between the new pixel value and the pixel average is compared to a set threshold to determine whether the new value is a foregorund or background.