Baku State University Journal of Mathematics & Computer Sciences ***2025, v. 2 (1), p. 46-56***

journal homepage: http://bsuj.bsu.edu.az/en

APPLICATION OF FUZZY LOGIC IN THE DYNAMIC SELECTION OF BACKGROUND SUBTRACTION METHODS

Vugar J. Gasimov*

Azerbaijan Technical University

Received 17 January 2025; accepted 21 February 2025

DOI: **https://doi.org/10.30546/209501.101.2025.3.201.017**

Abstract

This article explores the use of fuzzy logic in the adaptive selection of background subtraction methods. Methods such as MOG2, KNN, and Simple Differencing are used for object detection. However, under dynamic backgrounds and changing illumination conditions, these methods do not provide optimal results. To address this problem, the article proposes a dynamic selection mechanism for various background subtraction methods using fuzzy logic, which enhances detection accuracy and ensures efficient resource utilization.

Keywords: background subtraction, fuzzy logic, dynamic environments, object detection
***Mathematics Subject Classification* (2020):** 68T10, 68U10, 94A08, 68Q32.

1. Introduction

Detecting and tracking moving objects is of great importance in various applications, including video surveillance, traffic management, robotics, and human-computer interaction. One of the most crucial steps in this process is the background subtraction method, which separates moving objects from the background.

There are various methods available for background subtraction. Traditionally, methods such as Mixture of Gaussians (MOG2), Running Gaussian Average, Kernel Density Estimation (KDE), and simple frame differencing are widely used. Each of these methods is effective in specific environments and conditions, but they also have certain limitations. For instance, simple methods may provide accurate results in scenes observed with static or fixed cameras, but for dynamic backgrounds, changing illumination, and complex object shapes, other methods are necessary.

This article explores the application of fuzzy logic to ensure the adaptive selection of various background subtraction methods within the same application framework. The main advantage of fuzzy logic is its ability to better model real-world uncertainties and nonlinear changes. Thus, fuzzy logic can select the most appropriate background subtraction method by considering factors such as the amount of movement, lighting levels, and the dynamic nature of the background. This approach ensures more accurate and efficient use of background subtraction methods across various environments and conditions [5,6].

2. Problem definition

The main objective of this study is to develop a dynamic selection mechanism for background subtraction methods using fuzzy logic and evaluate its performance. Such an approach will not only improve the accuracy of object detection in various environments but also contribute to the optimal use of resources. In the remainder of the article, the background subtraction methods, the application of fuzzy logic, and the details of the proposed adaptive model are explained, followed by a discussion of the results and future perspectives.

Although each method used in background subtraction algorithms is effective for specific situations and environments, it is necessary to adaptively select different methods to achieve accurate results in dynamic environments. In this work, a fuzzy logic-based approach is proposed to ensure the dynamic selection of background subtraction methods.

3. Background Subtraction Methods

In this study, three main background subtraction methods are applied:

- **Simple Frame Differencing.** Simple Frame Differencing is the simplest background subtraction method and is primarily based on the differences between two consecutive frames. It is mainly used in environments with a static background and considers the changing pixels as moving objects [3,4]. Operating Principle: Two consecutive frames are taken, and the difference between them is calculated for each pixel. If the calculated difference exceeds a certain threshold, that pixel is considered part of a moving object. The differentiated pixels are used as a mask to subtract the background. It is simple and very fast. It requires minimal computational power.
- **Mixture of Gaussians (MOG2).** Mixture of Gaussians (MOG2) is a more complex and flexible background subtraction method. This method applies a Mixture of Gaussians to model the different states of each pixel. As a result, it provides more accurate results in dynamic backgrounds and changing illumination conditions [1,2]. Operating Principle: Gaussian models are created for each pixel. These models are updated over time, distinguishing between background and movement patterns. As the background changes over time, the models are updated, allowing better handling of lighting changes and other dynamic background elements. When pixel values do not match the models, that pixel is considered part of a moving object, and a mask is generated. It is effective in dynamic background conditions and changes in illumination. It adapts to changing environments by updating the background over time. It provides more accurate results in scenes with complex backgrounds and objects. It requires more computational resources compared to simple frame differencing.
- **KNN (K-Nearest Neighbors).** The KNN-based background subtraction method relies on the statistical analysis of each pixel's recent values and helps detect objects in complex scenes [4]. Operating Principle: The pixel values observed over previous frames are stored for each pixel. For each new pixel value, the similarity between this value and the past pixel values is calculated. If the

pixel's value matches the past pixel values, it is considered part of the background; otherwise, it is considered a moving object. It accurately models the background based on long-term observations. It is more resilient to noise and false positives.

The effectiveness of each of these methods is related to factors such as the amount of movement, changes in lighting, and the complexity of the background. Thus, fuzzy logic is applied to dynamically select the most appropriate background subtraction method by considering these factors [5].

4. Application of fuzzy logic

Fuzzy logic allows for more flexible decision-making by incorporating uncertain and nonlinear factors into the decision-making process. In cases where the amount of movement is slightly high or low, or when lighting is variable, making precise decisions with classical methods can be challenging. In this study, the fuzzy logic system takes the amount of movement as an input variable and, based on this information, selects the appropriate background subtraction method [5,7].

Components of the Fuzzy Logic System:

- Input Variable: Amount of movement (based on the number of differing pixels between frames).
- Output Variable: Background subtraction method (Simple Differencing, MOG2, KNN).
- Fuzzy Sets: The amount of movement is divided into fuzzy sets such as "low", "medium" and "high**".** The method selection is defined as "simple" (simple differencing), "mog2" and "knn".
- Rules:
    - When the amount of movement is "low," Simple Differencing is applied.
    - When the amount of movement is "medium," KNN is applied.
    - When the amount of movement is "high," MOG2 is applied.

Below is a simple Python code example for dynamic selection of background subtraction methods using fuzzy logic:

```
import cv2
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl

# Create background subtraction methods
mog2_subtractor = cv2.createBackgroundSubtractorMOG2()
knn_subtractor = cv2.createBackgroundSubtractorKNN()
simple_subtractor = None  # For simple frame differencing, storing the previous
frame

# Open the video stream
cap = cv2.VideoCapture('video_path.mp4')

prev_frame = None  # To store the previous frame for frame differencing

# Create input and output variables for the fuzzy logic system
movement_amount = ctrl.Antecedent(np.arange(0, 10001, 1),
'movement_amount')
method_selection = ctrl.Consequent(np.arange(0, 3, 1), 'method_selection')

# Fuzzy sets for movement amount
movement_amount['low'] = fuzz.trimf(movement_amount.universe, [0, 0, 3000])
movement_amount['medium'] = fuzz.trimf(movement_amount.universe, [2000,
5000, 8000])
movement_amount['high'] = fuzz.trimf(movement_amount.universe, [5000,
10000, 10000])

# Fuzzy sets for method selection
method_selection['simple'] = fuzz.trimf(method_selection.universe, [0, 0, 1])
```

```
    method_selection['knn'] = fuzz.trimf(method_selection.universe, [0, 1, 2])
    method_selection['mog2'] = fuzz.trimf(method_selection.universe, [1, 2, 2])

    # Define fuzzy rules
    rule1 = ctrl.Rule(movement_amount['low'], method_selection['simple'])
    rule2 = ctrl.Rule(movement_amount['medium'], method_selection['knn'])
    rule3 = ctrl.Rule(movement_amount['high'], method_selection['mog2'])

    # Create the fuzzy control system
    method_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
    method_simulation = ctrl.ControlSystemSimulation(method_ctrl)

    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break

        if prev_frame is not None:
            # Calculate the difference between frames to measure the amount of
movement
            diff = cv2.absdiff(prev_frame, frame)
            non_zero_count = cv2.countNonZero(cv2.cvtColor(diff,
cv2.COLOR_BGR2GRAY))

            # Make a decision using the fuzzy logic system
            method_simulation.input['movement_amount'] = non_zero_count
            method_simulation.compute()

            # Round the fuzzy output to select a method
            method_choice =
int(round(method_simulation.output['method_selection']))
```

```
        if method_choice == 0:  # Simple frame differencing
            if simple_subtractor is None:
                simple_subtractor = prev_frame.copy()
            fg_mask = cv2.absdiff(simple_subtractor, frame)
            fg_mask = cv2.cvtColor(fg_mask, cv2.COLOR_BGR2GRAY)
            _, fg_mask = cv2.threshold(fg_mask, 50, 255, cv2.THRESH_BINARY)
            method = "Simple Frame Differencing"
        elif method_choice == 1:  # Use KNN
            fg_mask = knn_subtractor.apply(frame)
            method = "KNN"
        elif method_choice == 2:  # Use MOG2
            fg_mask = mog2_subtractor.apply(frame)
            method = "MOG2"

        # Display the results
        cv2.putText(frame, f"Using: {method}", (10, 30),
  cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
        cv2.imshow('Foreground Mask', fg_mask)
        cv2.imshow('Original Frame', frame)

        # Wait for 'q' key to exit
        if cv2.waitKey(30) & 0xFF == ord('q'):
            break

    # Update the previous frame
    prev_frame = frame.copy()

# Release video and close windows
cap.release()
cv2.destroyAllWindows()
```

In this study, the application of fuzzy logic for the dynamic selection of background subtraction methods was explored, and the process of adaptively selecting different background subtraction methods was implemented. The proposed fuzzy logic-based system uses the amount of movement as an input variable and selects the appropriate background subtraction method based on this information. Based on fuzzy rules, the amount of movement is classified into different levels, and the most appropriate background subtraction method is applied for each level.

Since the amount of movement is classified into different levels in the fuzzy logic system, the selection of background subtraction methods becomes more precise. This approach not only simplifies the background and object detection process but also ensures high accuracy, as the most suitable background subtraction method is dynamically assigned for each scene.

The fuzzy logic system can utilize various parameters to select background subtraction methods in video. To transform the system into a multi-parameter fuzzy logic model, we can consider additional factors that may influence the selection of background subtraction methods, such as lighting conditions, noise levels, and object size. When parameters such as lighting conditions, scene complexity, object speed, camera stability, noise level, and background size are added, the fuzzy system can make more precise decisions when selecting a background subtraction method. Here is an example of a background subtraction method using a multi-parameter fuzzy logic system:

```
# Adding fuzzy variables
lighting_variation = ctrl.Antecedent(np.arange(0, 101, 1), 'lighting_variation')
scene_complexity = ctrl.Antecedent(np.arange(0, 101, 1), 'scene_complexity')
object_speed = ctrl.Antecedent(np.arange(0, 101, 1), 'object_speed')

# Defining fuzzy sets for these parameters
lighting_variation['low'] = fuzz.trimf(lighting_variation.universe, [0, 0, 30])
```

```
lighting_variation['medium'] = fuzz.trimf(lighting_variation.universe, [20, 50, 80])
lighting_variation['high'] = fuzz.trimf(lighting_variation.universe, [70, 100, 100])
scene_complexity['simple'] = fuzz.trimf(scene_complexity.universe, [0, 0, 30])
scene_complexity['moderate'] = fuzz.trimf(scene_complexity.universe, [20, 50, 80])
scene_complexity['complex'] = fuzz.trimf(scene_complexity.universe, [70, 100, 100])
object_speed['slow'] = fuzz.trimf(object_speed.universe, [0, 0, 30])
object_speed['medium'] = fuzz.trimf(object_speed.universe, [20, 50, 80])
object_speed['fast'] = fuzz.trimf(object_speed.universe, [70, 100, 100])

# Adding rules that incorporate these parameters
rule4 = ctrl.Rule(lighting_variation['high'] & scene_complexity['complex'], method_selection['mog2'])
rule5 = ctrl.Rule(object_speed['fast'] & movement_amount['medium'], method_selection['knn'])
rule6 = ctrl.Rule(scene_complexity['simple'] & movement_amount['low'], method_selection['simple'])

# Updating the control system
method_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6])
method_simulation = ctrl.ControlSystemSimulation(method_ctrl)
```

Selecting background subtraction methods based on the amount of movement allows the system to use more complex and resource-intensive methods only when necessary. In stable environments, simpler methods are used, reducing computational power and increasing real-time processing speed. In dynamic environments, when there is a lot of movement, more complex methods (MOG2, KNN) are selected [7,9,10]. Thus, computational resources are used only when needed, ensuring efficient resource management. This is a significant advantage, especially for systems with limited resources, such as mobile devices and security cameras.

5. Conclusion

The results show that the adaptive selection of background subtraction methods using a fuzzy logic system enhances overall performance. When the MOG2 method is selected in cases of high movement, the background subtraction process becomes more efficient and accurate. When the movement amount is low, selecting the simple frame differencing method conserves computational resources and provides faster results.

As a continuation of this study, further research can be conducted in this field, and new directions can be identified for expanding the system. The background subtraction method can be further enhanced by applying a multi-parameter fuzzy system, combining fuzzy logic with deep learning, and utilizing approaches such as interactive background subtraction.

References

[**1**] KaewTraKulPong P., Bowden R. An Improved Adaptive Background Mixture Model for Real-Time Tracking with Shadow Detection. European Workshop on Advanced Video-Based Surveillance Systems, 2001.

[**2**] Bouwmans T., El Baf F., Vachon B. Background Modeling Using Mixture of Gaussians for Foreground Detection—A Survey. Recent Patents on Computer Science, 2008, Vol. 2, No. 3, pp. 219–237.

[**3**] Szeliski R. Computer Vision: Algorithms and Applications. Springer-Verlag, 2011. https://doi.org/10.1007/978-1-84882-935-0

[**4**] Bouwmans T. Traditional and Recent Approaches in Background Modeling for Foreground Detection: An Overview. Computer Science Review, 2014, Vol. 11-12, pp. 31-66. https://doi.org/10.1016/j.cosrev.2014.04.001

[**5**] Mendel J. M. Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice Hall, 2001.

[**6**] Haralick R.M., Shapiro L.G. Computer and Robot Vision (Vol. 1). Addison-Wesley, 1992.

journal homepage: http://bsuj.bsu.edu.az/en
[**7**] Haladjian R., Sojka P. Adaptive Thresholding for Background Extraction. Proceedings of the 12th International Conference on Computer Analysis of Images and Patterns, 2014.

[**8**] Acharya T., Ray A. Image Processing: Principles and Applications. New York: Wiley, 2005.

[**9**] Stauffer C., Grimson W.E.L. Adaptive background mixture models for real-time tracking. Computer Vision and Pattern Recognition, IEEE, 1999. https://doi.org/10.1109/CVPR.1999.784637