*Baku State University Journal of Mathematics &Computer Sciences* ***2024****, v.1 (3), p 10-20*

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

# APPLICATION OF MULTI-KERNEL APPROACH IN BACKGROUND SUBTRACTION METHOD UNDER DYNAMIC ILLUMINATION CONDITIONS

Vugar J. Gasimov*

*Baku State University, Baku, Azerbaijan*

*Received 25 June 2024; accepted 17 September 2024*

[***https://doi.org/10.30546/209501.101.2024.1.3.020***](https://doi.org/10.30546/209501.101.2024.1.3.020)

---

**Abstract**

This study addresses the application of different kernel sizes and shapes to optimize morphological operations in the process of background subtraction in video. In this approach, a multi-kernel strategy is employed to adapt to various environmental conditions, such as illumination and object sizes. In this way, the accuracy of object detection is enhanced by adaptively adjusting the morphological operations using a multi-kernel strategy to refine the results of background subtraction.

*Keywords:* multi-kernel approach, background subtraction, morphological operations, object detection, dynamic kernel selection.

*Mathematics Subject Classification* (2020): 68T10, 68U10, 94A08.

---

## 1. Introduction

The background subtraction algorithm is one of the fundamental classical methods for object detection and tracking in computer vision. This method is widely used in applications where the detection and tracking of moving objects are crucial, such as surveillance, traffic monitoring, and quality control. The background subtraction process is based on the principle of distinguishing foreground objects (moving objects) in the video from static or slow-moving backgrounds. The effectiveness of this method is closely related to the precise modeling and updating of the background.

The main challenges encountered in background subtraction include dynamic changes in illumination, weather conditions, camera movement, and noise in the video. These factors can lead to false object detection or a decrease in detection quality. To address this problem, morphological operations, particularly erosion and dilation, play a significant role. Erosion narrows the object's boundaries, reducing noise, while dilation strengthens the structure of objects and fills in gaps. The correct application of these operations can enhance detection quality, significantly improving the accuracy of subsequent tasks such as object tracking and activity analysis [2, 5].

The multi-kernel approach further optimizes this process by allowing the application of morphological transformations tailored to different scenes and object characteristics [2, 5].

## 2. Problem definition

Accurate modelling of the background is a key requirement for the effectiveness of the background subtraction technique. In this study, the MOG2 (Mixture of Gaussians) method was used for the background subtraction process. The MOG2 model adaptively updates the background, allowing it to adapt to dynamic changes and distinguish moving objects from background elements [5].

Morphological operations such as erosion and dilation were utilized in the analysis of the videos. Erosion removes small noise and narrows the object's boundaries, while dilation fills gaps in the objects, preserving their integrity. Traditionally, a single kernel size and shape is used for these operations. However, this approach may not always be optimal for varying object sizes and scene conditions [2]. The multi-kernel approach creates a feedback mechanism that dynamically selects the size and shape of the kernel within video frames, optimizing the object detection process [2, 5].

## 3. Application of the multi-kernel approach

The multi-kernel approach allows for the selection of kernels tailored to various scene and object characteristics. In this study, circular, rectangular, and diamond-shaped kernels were used. Each kernel is applied according to its specific purpose. Small kernels are used to detect small objects, while large kernels are employed to preserve the contours of larger objects. In dynamic conditions (e.g., changes in illumination or varying object sizes), the kernel sizes are adaptively adjusted [3].

The selection of kernels is based on the object size and illumination conditions. Small kernels are applied for small objects and low-light scenes, while large kernels are used for large objects and highly illuminated scenes. The selection of the kernel is refined through a feedback mechanism based on the results obtained by the background subtraction algorithm in previous frames [7].

In this study, the following OpenCV-based Python code was used for applying background subtraction and morphological operations in videos. In the simple example below, dynamic kernel selection is performed based on varying illumination conditions and object sizes:

```
import cv2
import numpy as np

# Video capture
cap = cv2.VideoCapture('videopath')

if not cap.isOpened():
    print("Error: Could not open video file.")
    exit()

# Create different kernels
small_kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3))
large_kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (7, 7))

# Background subtractor
back_sub = cv2.createBackgroundSubtractorMOG2()

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

```
    # Display original frame for debugging
    cv2.imshow('Original Frame', frame)

    # Calculate the average brightness of the frame
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    average_brightness = np.mean(gray_frame)

    # Debug print
    print("Brightness:", average_brightness)

    # Set scene conditions based on brightness
    scene_conditions = {}
    if average_brightness < 50:  # Adjust this threshold based on your video
        scene_conditions['low_light'] = True
    else:
        scene_conditions['low_light'] = False

    # Apply background subtraction
    fg_mask = back_sub.apply(frame)

    # Estimate the object size by the number of white pixels in the foreground
mask
    num_white_pixels = np.sum(fg_mask == 255)
    print("White pixels (object size):", num_white_pixels)

    # Choose the kernel size based on the environmental conditions or scene
complexity
    if scene_conditions.get('low_light'):
        kernel = large_kernel
    else:
        kernel = small_kernel

    # Further adjust kernel size based on estimated object size
    if num_white_pixels > 5000:  # If object size is large (adjust threshold as
needed)
        kernel = large_kernel
    else:
```

```
        kernel = small_kernel

    # Morphological opening to remove noise
    fg_mask = cv2.morphologyEx(fg_mask, cv2.MORPH_OPEN, kernel)

    # Display result
    cv2.imshow('Foreground mask', fg_mask)
    if cv2.waitKey(25) & 0xFF == ord('q'):
        break

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

This code snippet demonstrates how background subtraction and morphological operations are performed, enhancing the object detection process in the video. The key challenge is finding an optimal threshold balance that accurately reflects the illumination conditions.

Conducting testing and fine-tuning is essential. You may want to test the program under various illumination conditions and visually inspect the results to determine the most effective threshold. Adjusting this based on the obtained empirical evidence will yield the best results [6].

Selecting the optimal threshold values to distinguish low-light and well-lit conditions depends on the specific characteristics of the camera used and the environmental lighting conditions [7]. The amount of noise in the video can increase based on the camera's sensitivity settings (ISO sensitivity).

Suppose we need to adjust this threshold for night and morning scenarios; we should consider the following nuances:

For night scenes, a lower threshold value is usually more suitable for effectively detecting low-light conditions. This can range between 20-40 on a scale of 0-255 (where 0 is completely black and 255 is completely white). You can start with a threshold of 30 and adjust the value based on the specific noise levels and detail requirements of your video.

During dawn or twilight, to adapt to changing light levels, you might need a slightly higher threshold compared to night. The initial threshold can range between 40-60. This helps manage the transition from low light to better illumination without losing important details due to excessive smoothing.

The ISO number determines the sensitivity of the camera's sensor to light. Lower ISO values (e.g., ISO 100 or 200) are less sensitive to light, resulting in a cleaner image with less noise. Higher ISO values (e.g., ISO 1600, 3200, or higher) increase the camera's sensitivity to light, allowing for brighter images in dark conditions. However, images captured with a high ISO setting have an increased amount of noise.

Since the amount of noise in the video varies with the ISO value, the threshold used for object detection can be adjusted by taking this parameter into account.

Let's look at the following simple code snippet for adjusting the threshold based on the ISO value:

```
import cv2
import numpy as np

# ISO value (this value can be obtained from the camera or metadata)
iso_value = 1600  # For example, 1600 is a high ISO value

# Adjust the threshold value based on the ISO value
# Apply a higher threshold for higher ISO values
if iso_value < 400:
    threshold_value = 50  # Lower threshold for low ISO
elif 400 <= iso_value <= 800:
    threshold_value = 70  # Medium threshold for medium ISO
else:
    threshold_value = 100  # Higher threshold for high ISO

# Apply thresholding for image binarization
ret, fg_mask = cv2.threshold(frame, threshold_value, 255, cv2.THRESH_BINARY)

# Display the result using the threshold value
cv2.imshow('Foreground mask', fg_mask)
```

Since the ISO value is a camera parameter, it can be obtained through the camera settings or video metadata. Here, the value (ISO 1600) is provided as an example. The ISO value is checked within specific ranges, and an appropriate threshold value is assigned accordingly. Since a high ISO (e.g., 1600 and above) causes more noise, the threshold value is increased. This will preserve only the more precise information in the objects and remove noisy pixels from the foreground. For each frame of the video, binarization is applied using the appropriate threshold value to detect objects.

In obtaining the results, factors such as object size, ISO parameter, and lighting conditions are considered, allowing the kernels to be dynamically adjusted. In these adjustments, the use of soft computing methods can further optimize processes such as background modeling and the selection of kernel sizes and shapes:

**Fuzzy logic**

Fuzzy logic enables more flexible decision-making in dynamic and uncertain environments. We can apply fuzzy logic to adjust the effects of variables such as object size, illumination conditions, and noise levels. For example, we take object size, illumination level, and noise amount as input variables, and the kernel size and shape as output variables. We formulate fuzzy rules such as: "If the illumination is low and the object size is large, then select a large kernel" or "If the object size is small and the illumination is good, then select a small kernel". These fuzzy rules will allow for more precise adjustments in object detection. Thus, we can use fuzzy logic in the code to change the size and shape of the morphological kernel.

**Artificial neural networks**

Artificial neural networks can be used to automate dynamic adjustments in object detection and background subtraction processes. Neural networks can learn the features of objects in video frames (boundaries, contours, etc.) and select appropriate kernel sizes for different objects. For this, the neural network is trained on video frames by applying various kernels. The neural network is trained to evaluate object size and noise level to select the appropriate kernel.

**Genetic algorithms**

Genetic algorithms can be used to find the optimal parameters for the sizes and shapes of various kernels. This approach can optimize the parameters of morphological operations (kernel size, shape, number of iterations, etc.) used to enhance the accuracy of background subtraction and object detection processes.

In the first step, an initial population describing the parameters (e.g., kernel sizes, shapes) is created. Then, each member of the population is evaluated using a target function (e.g., object detection accuracy, noise reduction). Through genetic operations (crossover, mutation), the parameters are evolved to approach the optimal configuration. Thus, genetic algorithms can find the most suitable parameters for various environmental conditions through testing and optimization.

**Adaptive thresholding methods**

Adaptive thresholding methods can be used to determine the boundaries of objects based on illumination conditions. Clustering methods like K-means can optimize kernel selection by grouping objects based on their varying sizes and characteristics.

## 4. Conclusion

The application of multi-kernel morphological operations in background subtraction has provided the following advantages:

1. Dynamically adjusted kernels adapt to changes in illumination and weather conditions, ensuring more accurate object detection. In low-light scenes, larger kernels have reduced noise, while in well-lit scenes, smaller kernels preserved details, maintaining more accurate object boundaries.

2. The size of the kernels was adjusted according to the size of the objects. Using small kernels for small objects helped in detecting the object with greater detail. The use of larger kernels for large objects helped preserve their contours and allowed for more complete detection.

3. As a result of the correct application of erosion and dilation operations, noise was significantly reduced. Small kernels eliminated fine noise, while large kernels filled gaps within objects, enhancing detection accuracy.

4. The dynamic selection of various kernels in response to changing conditions in video streams ensured more reliable object detection.

The results indicate that multi-kernel morphological operations significantly refine the detection and tracking process, adapting to various object characteristics and scene conditions. In traditional single or limited kernel approaches, the lack of consideration for varying object sizes, movement characteristics, and environmental conditions can lead to incorrect detection results or incomplete object detection. In such cases, accurately detecting the boundaries of objects becomes challenging, and the separation of the foreground from noise may be insufficient. The multi-kernel approach addresses this problem by offering a flexible processing strategy that adaptively adjusts to object sizes and scene characteristics. This method has demonstrated high effectiveness in dynamic conditions such as surveillance, traffic monitoring, and environmental observation.

Future research can focus on further enhancing the multi-kernel approach by utilizing deep learning and other advanced soft computing methods in background modeling. This could provide higher accuracy and efficiency in the broad application areas of background subtraction technology.

## References

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

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

[3] Toyama K., Krumm J., Brumitt B., Meyers B. Wallflower: Principles and practice of background maintenance. Proceedings of the 7th IEEE International Conference on Computer Vision. 1999, p. 255-261

[4] Stauffer C., Grimson W.E.L. Adaptive background mixture models for real-time tracking. Computer Vision and Pattern Recognition, IEEE, 1999.

[5] Zivkovic Z. Improved adaptive Gaussian mixture model for background subtraction. International Conference on Pattern Recognition, IEEE, 2004.

[6] Kaehler A., Bradski G. Learning OpenCV 4: Computer Vision with Python. O'Reilly Media, 2019.

[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] Serra J. Image Analysis and Mathematical Morphology. New York: Academic Press, 1983.

[9] Soille P. Morphological Image Analysis: Principles and Applications. Springer, 2003.