Background Subtraction

*

Tanmaya Karmarkar
Computer Science, CMPS, I. K. Barber Faculty of Science
UBC Okanagan
Kelowna, Canada
tanmayak@student.ubc.ca

Karel Joshua Harjono
Computer Science, CMPS,S I. K. Barber Faculty of Science
UBC Okanagan
Kelowna, Canada
harjono@student.ubc.ca

Stephen Lee
Mechanical Engineering, MECH, School of Engineering
UBC Okanagan
Kelowna, Canada
Stlee9@student.ubc.ca

Trevor Winser
Computer Science, CMPS, I. K. Barber Faculty of Science
UBC Okanagan
Kelowna, Canada
trevorwinserschool@gmail.com

***Abstract*—This paper presents a review and study on background subtraction techniques for volleyball player and ball detection. We explore methodologies such as MoG, MoG2, KNN and GMG, evaluating their performance in four videos of volleyball games. We evaluate its performance on four volleyball game videos, demonstrating its computational efficiency and effectiveness under controlled conditions. However, we highlight trade-offs between simplicity and robustness, particularly when compared to deep learning models. While our approach provides lightweight deployment advantages, it is sensitive to manually defined thresholds and lacks adaptability to dynamic environments. Despite its limitations, our system serves as a strong baseline for volleyball tracking and a valuable preprocessing tool for more complex pipelines in sports analytics.**

***Index Terms*—Background Subtraction, BGS, GMM, Volleyball, MoG**

I. Introduction

Background subtraction (BGS) is a fundamental technique in computer vision that is used to separate foreground objects from the background in images or videos. It is commonly used to improve object detection, especially for small and moving objects. Although there are a multitude of applications for BGS, from hydrocarbon leak detection [1] to traffic surveillance [2], the area under investigation in this article is volleyball detection.

The use of computer vision in sports analytics has increased significantly, enhancing performance assessment and game officiating. Applications such as player tracking, motion analysis, automated referee assistance, and ball trajectory estimation enable real-time statistics generation, coaching support, and post-game analysis. One well-known application is the Hawk-Eye system, which is used in sports such as tennis, cricket, and volleyball to track balls in and out of play and create animations of the movements of the detected balls [3]. Among these applications, however, ball tracking remains a challenging problem due to the high-speed nature of play, frequent occlusions, and environmental factors such as illumination variations and background clutter.

The primary objective of this project was to develop a robust volleyball tracking system by leveraging background subtraction techniques. This was further extended to also include player detection. Accurately tracking the volleyball's movement allows for improved detection of shot types and provides deeper insights into game strategies. By overcoming challenges such as occlusions, rapid motion, and dynamic backgrounds, this research aims to enhance player performance evaluation, coaching decisions, and referee accuracy in live matches. Furthermore, this project contributes to the broader field of sports analytics by refining background subtraction for fast-paced and dynamic sports environments.

II. Literary Review

Background subtraction (BGS) is a fundamental technique in computer vision, particularly for applications like motion detection, surveillance, and object tracking. Traditional background subtraction has been extensively researched and continues to evolve due to its wide-ranging applications. BGS techniques typically consist of four main steps, which are:

1) **Background Modeling:** Establishing a model that represents the background, distinguishing between unimodal and multimodal backgrounds.
2) **Background Initialization:** Computing the initial background image used as a reference for detecting changes over time.
3) **Classification of Pixels:** Classifying pixels as either foreground or background using statistical or machine learning methods.
4) **Background Maintenance:** Continuously updating the background model to adapt to environmental changes such as lighting variations and dynamic backgrounds.

Among the various BGS techniques, the **Mixture of Gaussians (MoG)** model has remained a widely adopted method.

Initially proposed by Stauffer and Grimson in 1999, the MoG model aims to avoid explicitly modeling pixel values while dealing robustly with lighting changes, repetitive motions, and slow-moving objects [4]. The MoG method models each pixel as a mixture of K Gaussian distributions. When a new pixel is introduced, the algorithm either matches it to an existing Gaussian or extends the variance of another distribution. This allows the model to classify dynamic objects, such as a swaying tree, without erroneously labeling it as background [4].

Despite its robustness, the MoG model has limitations in terms of adaptability. To address this, MoG2 was introduced by Zivkovic in 2004 [5]. MoG2 differs from its predecessor in that it uses an online method to determine the number of Gaussian distributions, as opposed to the fixed number of distributions in MoG. This enhancement allows MoG2 to better adapt to changes in the scene, such as varying light conditions, more complex backgrounds, and moving objects. The adaptability and improved performance of MoG2 make it an ideal algorithm for a variety of dynamic environments.

Both MoG and MoG2 are classified as **Gaussian Mixture Models (GMMs)**, which are a type of fuzzy model that offers several advantages over traditional binary classification methods. Rather than assigning each pixel to either foreground or background (i.e., binary classification), GMMs allow for a fuzzy, probabilistic classification, which enables the model to handle more complex scenarios, such as partial occlusions or moving objects blending into the background. Additionally, GMMs can accommodate multi-modal background distributions, a common feature in dynamic scenes [6].

In recent years, researchers have extended the use of GMMs by incorporating **fuzzy background subtraction models**. Instead of relying solely on binary (crisp) foreground/background designations, fuzzy models use multiple distributions (multi-modal) to more accurately detect foreground objects. The fuzzy GMM approach offers several advantages over traditional models by more effectively handling dynamic and changing backgrounds [7]. Fuzzy GMMs can model complex background scenes more robustly, as demonstrated by their superior performance in dealing with shadows and illumination changes, which often hinder simple background subtraction methods [8].

Additionally, various **feature extraction methods** have been explored to improve the robustness of BGS models. Color features have been widely used due to their effectiveness in differentiating objects from the background. However, texture, stereo, and edge-based features have also been integrated into the BGS process [9], allowing for more precise and stable foreground detection. The combination of these features can be further enhanced through advanced mathematical models such as the **Sugeno** and **Choquet integrals**, which enable more robust integration of multiple feature types and better handling of complex environmental changes, such as lighting variations or shadow effects [10].

In summary, while traditional background subtraction methods like MoG have been instrumental in many computer vision applications, there remains significant room for improvement. Advances in fuzzy Gaussian mixture models (fuzzy GMMs) offer more robustness to environmental changes, including complex dynamic backgrounds and illumination variations. Additionally, integrating additional feature types and leveraging advanced mathematical models can further improve the accuracy and adaptability of background subtraction techniques. The continued research and development of these methods are crucial for achieving reliable, real-time foreground detection in increasingly dynamic environments.

III. System Design and Implementation

This section presents a comprehensive explanation of the design, architecture, and implementation of the background subtraction system developed for analyzing volleyball matches. The primary objective of the system is to detect and visually highlight dynamic objects—mainly volleyball players and the ball—using traditional computer vision techniques. To achieve this, the system employs a combination of background subtraction algorithms to effectively segment moving objects from the static background.

At the core of the system lies the adaptive capabilities of the `BackgroundSubtractorMOG2` algorithm, which robustly models the static background while distinguishing moving foreground objects. The MOG2 algorithm is especially effective in handling illumination changes and slow-moving objects, making it ideal for capturing the dynamic nature of a volleyball match. Once motion components are extracted, they are refined using a series of morphological operations, such as dilation and erosion, which help remove noise and enhance the foreground detection.

In addition to MOG2, the system incorporates the `K-Nearest Neighbors` (KNN) algorithm for background subtraction. The KNN-based approach is utilized for its simplicity and efficiency in handling background updates over time. Unlike MOG2, which relies on Gaussian Mixture Models, KNN builds a history of pixel values and classifies each new pixel based on the majority of neighboring pixels. This method is particularly effective for capturing fast-moving objects and handling scenes with complex dynamic backgrounds, such as those encountered in sports video analysis. KNN's ability to update the background model incrementally allows it to adapt to rapid scene changes, making it suitable for real-time background subtraction.

Moreover, the detection of the volleyball is further enhanced using HSV (Hue, Saturation, Value) color space filtering techniques. By converting the image to HSV space, the system isolates the ball based on its distinctive color (yellow or blue) and reduces the effect of lighting variations. This color-based segmentation works synergistically with the motion-based methods to reliably track the ball's movement throughout the match.

Together, the combination of motion-based (MOG2, KNN) and color-based (HSV filtering) techniques ensures a high-quality visual output for motion segmentation in sports video. This hybrid approach allows for accurate tracking of players

and the volleyball, even in challenging environments with varying lighting conditions and complex motion.

A. System Architecture

The core input to our system is a pre-recorded video clip of a volleyball match, which is filmed using a stationary camera setup. This fixed viewpoint simplifies the process of modelling the background since the non-moving elements (such as the court floor, net, boundary lines, and surrounding environment) do not change significantly throughout the video. Given this static nature of the background, background subtraction becomes a suitable and highly effective technique for detecting motion—capturing both players and the ball as they move across the court.

The system is designed to process each frame sequentially, enabling real-time or near-real-time operation. It reads video frames, applies a background subtraction model to identify motion, filters and refines this motion mask using morphological operations to reduce noise, and then further processes the frames to detect and highlight the volleyball using color thresholding in the HSV color space.

System Inputs:

- A video file of a volleyball match, typically in .mp4 format.
- User-defined parameters for the background subtraction model MOG2 (e.g., learning rate, history length).
- Predefined HSV color thresholds specific to the volleyball's hue, saturation, and brightness values.

System Outputs:

- A video with the moving players and volleyball detected and highlighted.

B. Movement Detection Algorithms

Various background subtraction techniques include

1) **Frame Differencing** – one of the simplest approaches to detect motion. The idea is to subtract one frame from another (usually the current frame from a previous or reference frame) and identify the pixels where significant differences occur.
2) **Running Average Method** – updates the background by blending the current frame with the previous background using a weighted average, allowing gradual adaptation to changes in the scene.
3) **Gaussian Mixture Model (GMM)** – a probabilistic approach that represents each pixel's intensity as a mixture of several Gaussian distributions.

In this work, we focus on the Gaussian Mixture Model (GMM). GMM is a probabilistic approach that models each pixel as a mixture of several Gaussian distributions. This allows it to handle variations in pixel intensity over time, making it effective for distinguishing between background and moving foreground objects. The core idea behind GMM is to learn and update these Gaussian components to adapt to the scene dynamics. This model basically relies on the following concepts

1) **Multiple Gaussians per Pixel** – Each pixel is modeled as a mixture of K Gaussian distributions rather than a single value. This allows it to adapt to changes like lighting variations, waving trees, or moving water.
2) **Weights** – Each Gaussian component has a weight, updated over time. Background pixels tend to belong to stable, high-weight Gaussians, while foreground objects introduce new, low-weight Gaussians.
3) **Foreground Detection** – A new pixel value is compared to existing Gaussians. If it matches a background distribution, it is classified as background; otherwise, it is considered foreground.
4) **Adaptability** – Over time, the model adapts by updating or replacing Gaussian components to account for gradual scene changes.

1) Why Multiple Gaussians?: A single Gaussian assumes that a pixel has a fixed intensity with only small variations (due to noise). In real-world scenarios, pixel intensity can fluctuate due to factors such as shadows, lighting changes, temporary occlusions by moving objects, and periodic variations in background elements like water waves. By utilizing multiple Gaussians, the model can more effectively capture and adapt to these changes.

How It Works For each pixel, GMM maintains K Gaussian distributions (typically 3 to 5). Each Gaussian is characterized by Mean (The expected pixel intensity), Variance (The spread of pixel values) and weights (The probability that a pixel belongs to this Gaussian). At every frame, the new pixel value is compared with the existing Gaussians:

1) Matching a Gaussian: If the pixel value is close to an existing Gaussian (within 2.5 standard deviations), that Gaussian is updated.
2) No Match: A new Gaussian is introduced, possibly replacing the least significant one.
3) Background Identification: The most stable Gaussians (higher weights and lower variance) represent the background, while others are classified as foreground.

2) Advantages: The Gaussian Mixture Model (GMM) is effective in handling dynamic backgrounds and can model repetitive motions such as moving trees and waves. It adapts to illumination changes over time, ensuring flexibility in varying lighting conditions. Additionally, GMM is robust against noise and small motions, making it a reliable approach for background subtraction in complex environments.

3) Disadvantages: The disadvantages of this system is that it is computationally intensive compared to simpler methods like frame differencing. Additionally, if not properly tuned, it may misclassify sudden fast-moving objects, leading to inaccuracies in detection.

C. Improving MOG2 Results with HSV filters

In sports video analysis, common methods detect objects with either motion (background subtraction) or by specific color/appearance cues (color filtering in HSV space). Each approach has its advantages, and often they can complement

each other. Motion-based detection (like background subtraction) flags anything that moves against the static background. This is very useful when the object of interest does not have a uniquely identifying color but is distinguishable by movement. In our volleyball scenario, background subtraction will automatically detect all moving objects – players, the ball, or even moving shadows – without needing to know their color in advance. This broad detection is advantageous because the volleyball and players may have similar or varying colors, but as long as they move, they will be extracted from the static court background. Motion-based methods are thus color-agnostic and resilient to changes in the object's appearance or the background's color. They also adapt to the environment: for example, if the floor or wall colors change under different lighting, the background model will eventually absorb those as background, still highlighting only new movements. A drawback, however, is that motion-based detection cannot detect players when they are perfectly still (e.g., players staying still during serves) because a stationary object becomes part of the "background" model. Also, background subtraction by itself does not distinguish which moving object is the ball versus a player – additional logic is needed to separate the ball from players once all motion regions are obtained (for instance, based on size, shape, color, or trajectory).

By contrast, color-based detection (such as HSV color thresholding) focuses on identifying objects of a known color. In an HSV filtering approach, a color filter is manually/automatically set to include or exclude specific objects, such as a volleyball, which is often bright orange or yellow / blue patterned, by detecting pixels in the frame that fall within that color range. The advantage of this approach is that it can detect the ball even if it is not moving, as long as its color is distinctive and within the specified range. It also inherently ignores other moving objects that do not match the color – for instance, players (with different jersey colors) would be ignored, and ideally only the ball is segmented out. Color filtering can be very fast, and for a uniformly colored object under stable lighting, it can be quite reliable. However, color-based methods have limitations in dynamic, realistic settings. They rely heavily on the assumption that the object's color is unique and consistent. If the background or players' uniforms contain similar colors to the ball, simple HSV thresholding can falsely detect those areas as ball candidates. In indoor volleyball, the court flooring or advertisement boards might share color tones with the ball, especially under varying indoor lighting, making color segmentation challenging. Lighting changes can also alter the perceived color: a ball might appear different in hue or brightness depending on shadows or camera exposure, potentially causing missed detections if the HSV range is not adjusted. In short, color-based detection is sensitive to illumination and color variations, and typically requires careful calibration of color ranges.

In our implementation, these two approaches are combined to leverage the strengths of each. First, we use background subtraction to find all moving objects and then apply an HSV filter within those moving regions to specifically confirm which moving blob is the ball or player. Then, we use color detection to propose ball candidates and use motion mask to eliminate blobs that are stationery. Throughout the course of this project, we encountered challenges associated with both techniques and worked to leverage their respective strengths, ultimately developing a hybrid approach that integrates the advantages of each method for effectively isolating players and the volleyball from a given match.

D. KNN in Background Subtraction

KNN stands for K-Nearest Neighbors. K-Nearest Neighbors (KNN) is a non-parametric, data-driven algorithm used in background subtraction, particularly in OpenCV's cv::BackgroundSubtractorKNN. It models each pixel's background by maintaining a history of pixel values and classifying new observations based on the nearest neighbors in this history. The method works by storing a set of the most recent pixel values and then determining whether a new pixel belongs to the background or foreground based on its similarity to these stored values. The classification is done using a distance metric, typically Euclidean distance, where a pixel is considered background if a sufficient number of its nearest neighbors belong to the background model. KNN is effective in handling dynamic backgrounds, illumination changes, and small repetitive movements, making it suitable for real-world applications such as surveillance and object tracking. Mathematically, the probability of a pixel being background is computed based on the fraction of neighbors within a predefined distance threshold, allowing adaptive background modeling.

E. BackgroundSubtractorGMG

BackgroundSubtractorGMG is an advanced background subtraction algorithm available in OpenCV. It combines statistical background image estimation with Bayesian inference for foreground detection.

1) Key Features:

- **Pixel-Wise History:** Uses a long-term pixel-wise statistical history to model the background.
- **Bayesian Segmentation:** Employs a probabilistic framework to classify pixels as foreground or background.
- **Motion Sensitivity:** More responsive to moving objects compared to other background subtractors.
- **Shadow Removal:** Capable of reducing shadows but may still be sensitive to sudden lighting changes.
- **Adaptability:** Effective in dynamic scenes but requires an initial training phase.

2) Comparison with MOG2 and KNN:

- **Faster Initial Adaptation:** GMG adapts quickly to a scene but needs a warm-up period.
- **Higher Sensitivity:** Detects fine motion details but can be more prone to noise.
- **Memory Usage:** Higher than MOG2 but comparable to KNN due to its history-based approach.

GMG is particularly useful in scenarios where precise motion segmentation is required, such as surveillance and

object tracking. However, due to its sensitivity, it might require additional post-processing to handle noise effectively.

F. Comparison of MOG2, GMG, and KNN for Background Subtraction

Background subtraction is a crucial technique in computer vision for detecting moving objects in video streams. OpenCV provides three widely used methods: Mixture of Gaussians (MOG2), BackgroundSubtractorGMG, and K-Nearest Neighbors (KNN). Each method has its strengths and is suited for different scenarios.

1) MOG2 (Mixture of Gaussians 2):

- Uses a Gaussian Mixture Model to model the background.
- Adapts dynamically to lighting changes and moving background elements.
- Provides shadow detection, reducing false positives.
- Suitable for real-time applications with moderate computational cost.
- Struggles in highly dynamic environments.

2) GMG (BackgroundSubtractorGMG):

- Uses a statistical background estimation combined with Bayesian inference.
- More sensitive to motion, making it effective for detecting small moving objects.
- Requires an initial training phase before becoming effective.
- More prone to noise, requiring post-processing for optimal results.
- Consumes more memory compared to MOG2.

3) KNN (K-Nearest Neighbors Background Subtractor):

- Uses a non-parametric KNN model to classify pixels as foreground or background.
- Adapts well to complex and dynamic backgrounds.
- Performs better in scenes with frequent background changes.
- Higher computational cost compared to MOG2, making it less efficient for real-time applications.
- Provides robust results but requires tuning of parameters like history size and distance threshold.

Feature	MOG2	GMG	KNN
Adaptability	Moderate	High	High
Shadow Detection	Yes	No	No
Sensitivity to Motion	Medium	High	High
Computational Cost	Low	Medium	High
Memory Usage	Low	High	High
Noise Sensitivity	Moderate	High	Low
Suitability for Dynamic Backgrounds	Moderate	Low	High

TABLE I
COMPARISON OF MOG2, GMG, AND KNN BACKGROUND SUBTRACTORS

4) Performance Comparison:

5) Conclusion: Each background subtraction method has advantages and drawbacks. MOG2 is a good general-purpose algorithm with shadow detection, making it suitable for moderate dynamic environments. GMG offers high sensitivity to motion but requires careful parameter tuning. KNN provides the best adaptability to changing backgrounds but comes with a higher computational cost. The choice of method depends on the application requirements, such as real-time performance, accuracy, and scene complexity.

G. Source Code Diagrams

The algorithm presented in the pseudocode 22 outlines a real-time processing pipeline for detecting and highlighting volleyball players and the ball in a match video. The input is a standard video file, and the output is a processed video with only the relevant moving elements — players and ball — highlighted. Initially, the video is loaded using OpenCV's VideoCapture utility, and properties such as frame rate, width, and height are extracted. For performance optimization, the frame dimensions are scaled down before processing begins. A MOG2-based background subtractor is used to segment moving foreground objects from the static background. This is followed by morphological operations (opening and dilation) to remove noise and enhance the shape of moving players.

In parallel, ball detection is achieved using color thresholding in the HSV color space. Since volleyballs in the dataset tend to appear yellow, an HSV color range is defined to isolate yellow pixels. The resulting binary mask is refined using morphological operations, and the largest contour is assumed to represent the ball. A circular region is drawn around this contour to create a dedicated mask for the ball.

Both the player mask and ball mask are then applied to the original frame using bitwise operations to extract relevant regions. These two masks are also combined to generate a unified visualization of player and ball activity. The processed frame is then displayed and written to an output video file. The loop continues until all frames are processed or the user interrupts the program by pressing the 'q' key. Finally, all resources are released and OpenCV windows are closed.

H. Technologies Used for The System

OpenCV's Background Subtractor MOG2: OpenCV provides the BackgroundSubtractorMOG2 class, an adaptive algorithm based on GMM, which handles varying illumination and dynamic backgrounds.

We decided to proceed with python and opencv2

1) BackgroundSubtractorMOG2 in OpenCV: BackgroundSubtractorMOG2 is OpenCV's improved GMM-based subtractor that implements Zivkovic's adaptive GMM algorithms [11]. Notably, Zivkovic's method automatically determines the optimal number of Gaussians for each pixel instead of using a fixed number. This adaptive approach means MOG2 can model simple backgrounds with just one Gaussian per pixel, or more complex backgrounds (e.g., waving tree leaves or gym lights flickering) with multiple Gaussians as needed. MOG2 also offers better adaptability to changing illumination conditions in the scene. For example, if the indoor volleyball court's lighting changes or flickers, or sunlight moves across the floor, MOG2 will gradually adjust the background model to these changes. An important practical feature in MOG2 is the option to detect

Algorithm 1 Background Subtraction and Ball Detection

Input: Volleyball match video
Output: Video with players and ball highlighted
Load video and get FPS, width, height
Scale dimensions and initialize video writer
Create background subtractor (MOG2) and define morph kernels
while frames available **do**
    Read and resize frame
    if invalid frame **then**
        Break
    end if
    Apply background subtraction and clean with morphology
    Convert to HSV and threshold yellow color for ball detection
    Refine color mask with morphological operations
    Find largest yellow contour and draw 'ball mask'
    Find top 6 largest moving contours and draw 'player mask'
    Extract players and ball from frame using bitwise AND
    Combine both masks with bitwise OR and display output
    if 'q' pressed **then**
        Break
    end if
end while
Release video objects and close windows

and mark shadows (usually as gray pixels in the output mask). Moving shadows cast by players can confuse simpler motion detectors, but MOG2 can identify probable shadows so that they are not counted as foreground objects, thereby improving the robustness of detection in scenes with strong lighting and shadows. Overall, Gaussian mixture models like MOG2 provide a powerful balance between sensitivity to moving objects and robustness to background noise, at the cost of more computation and parameters compared to simpler methods.

While implementing our application we experimented with a few parameters and settings to enhance the performance of our output.

Experiments

1) While implementing our application, we experimented with several parameters and processing techniques to enhance the accuracy and visual clarity of player and ball segmentation. The use of a stationary camera allowed us to take advantage of temporal consistency across frames, and we explored both static and adaptive background modelling approaches.
   For background subtraction, we primarily used MOG2 and also experimented with KNN. One key parameter we tuned was the history length, which controls how many past frames are used to model the background. A smaller history made the model adapt quickly to scene changes, but this led to instability and flickering in the mask, especially when players paused momentarily during serves. Increasing the history smoothed out such fluctuations but made it slower to react to genuine foreground movement, causing some background to be misclassified as foreground. For MOG2, setting the history to around 500 frames yielded a good balance for our static scene.
2) We also investigated the varThreshold in MOG2 and dist2Threshold in KNN, both of which influence how sensitive the model is to changes in pixel intensity. Lower thresholds caused noise and small shadows to be classified as foreground, while higher values suppressed legitimate motion. We found that setting varThreshold=100 in MOG2 and dist2Threshold=500 in KNN helped suppress minor background motion while retaining the players' movement.
3) The shadow detection feature in both MOG2 and KNN was another important variable. Enabling shadow detection helped prevent large shadow regions from being classified as players. However, shadows often introduced mid-level pixel values (127) that needed to be explicitly removed to clean up the mask. We handled this by setting all pixels with value 127 to 0 before applying morphological operations.
4) In terms of morphological processing, we experimented with different kernel sizes and operations like opening, closing, and dilation. Smaller kernels helped preserve finer details (e.g., limbs, ball outlines), while larger kernels removed more noise but at the risk of erasing valid foreground objects. We found that an elliptical kernel of size (2, 2) for opening followed by a rectangular kernel of (8, 12) for dilation effectively retained the player shapes while reducing background clutter. These shapes are chosen based on the assumption that the players will most likely be standing still, so a longer vertical component helps to connect the separated blobs more effectively.
5) For color-based ball detection, we tested multiple HSV ranges to detect the yellow volleyball reliably. The range [10, 0, 0] to [50, 255, 255] was selected after observing that it captured the ball well under various lighting conditions without including unrelated yellowish objects. Morphological closing followed by opening was used to fill gaps from the color mask. We also tested alternative color spaces like RGB and Lab, but HSV provided the most robust results under shadows and lighting variation.
6) Lastly, we tested contour filtering by selecting the largest N contours in the movement mask, where N controls how many moving entities are retained (players or large objects). Increasing this number introduced more noise (e.g., referees, net movement), while too small a value risked missing one or more players. We found keeping the top 6 largest contours provided a good balance in our test clips, usually capturing all active players.

These experiments helped refine the pipeline to work reliably across different video segments, enabling accurate tracking and segmentation of volleyball gameplay.

IV. Results and Visualization

To evaluate the performance of our object segmentation and tracking pipeline, we constructed a four visualization masks from the video frames: the original frame, a "Players Only" frame, a "Ball Only" frame, and a combined "Ball + Players" view. This layout allows us to examine each component of our system both independently and in combination, offering insight into how well our processing pipeline isolates key dynamic elements from the video.

The **Original** frame, Figure 1[a] 2[a], which is the unmodified RGB image of the current video frame (resized for performance). This frame includes all elements in the court environment, including the court lines, net, audience, the referee, and any camera operators. It provides a reference for evaluating the success of our detection pipeline in removing background clutter and isolating meaningful motion.

The **Players Only** frame, Figure 1[b], which is generated by applying the MOG2-based background subtraction followed by morphological filtering and a contour-based selection of the six largest regions of motion. In general, this frame accurately captures the moving volleyball players. However, we observed that static or slow-moving individuals such as the referee and occasionally the cameraman were also captured in the foreground mask. This occurs because they are not part of the background model, and their presence remains relatively static but still differs slightly from the background (e.g., subtle movements or differing clothing texture), which leads the subtractor to classify them as moving foreground. In our results, the referee is often incorrectly highlighted as a foreground object, particularly when standing behind the players or near the edge of the frame. This constitutes a misdetection and reveals a limitation of contour filtering solely based on size - as the referee shadows often has an area comparable to a player.

The **Ball Only** frame, Figure 1[c], which highlights the yellow volleyball based on color segmentation in the HSV color space. The defined range targets hues typical of a standard volleyball and is effective in isolating the ball under normal lighting conditions. The largest yellow contour is selected and enclosed with a circle to create a binary ball mask. While generally robust, this approach is sensitive to lighting and color similarity. For instance, parts of the referee's shirt or banners near the court can occasionally match the yellow threshold, especially if the ball is blurred during fast motion. These can result in false positives, although morphological cleaning steps (opening and closing) help reduce noise.

Finally, the **Ball + Players** view, Figure 1[d], which is created by combining the two previous binary masks using a bitwise OR operation. This frame showcases the cumulative result of our segmentation pipeline: players and the ball are clearly visualized while the background is effectively removed. This output is particularly useful for downstream applications such as player tracking, game analytics, or training computer vision models. However, the inclusion of non-player figures like the referee in this composite frame reminds us of the need for additional semantic filtering (e.g., pose estimation, jersey detection, or bounding box aspect ratio analysis) to more precisely distinguish between players and static bystanders.

Through this multi-frame visualization, we gain a comprehensive understanding of how the system processes each frame — from raw video to isolated movement. It also enables us to identify key strengths and weaknesses, such as the system's sensitivity to motion thresholds, reliance on color segmentation for ball detection, and current limitations in distinguishing true foreground from semi-static elements like referees or camera operators. These findings inform future improvements in our approach, such as refining contour filtering heuristics, incorporating background freezing techniques, or integrating higher-level semantic cues to reduce misdetections.

V. Discussions

Our system demonstrates a practical and effective approach for detecting players and tracking the volleyball in match footage using traditional computer vision techniques, primarily background subtraction and color segmentation. This section highlights the strengths and limitations of the pipeline, discusses observed results and anomalies, and compares its performance with more advanced or alternative methods.

A. Advantages

One of the key strengths of our system lies in its simplicity and real-time capability. The use of OpenCV's built-in background subtraction algorithms, such as KNN and MOG2, allows for fast and reasonably accurate segmentation of moving objects without requiring any prior training. The pipeline is fully interpretable and tunable, making it easy to understand and adapt to different camera angles or lighting conditions.

Additionally, the use of HSV color thresholding for ball detection proved effective under consistent lighting, particularly since the volleyball used in the footage had a distinct yellow color. The modular structure of the code also allows for easy experimentation with different components, such as changing the background model, tuning morphological parameters, or switching to different color spaces.

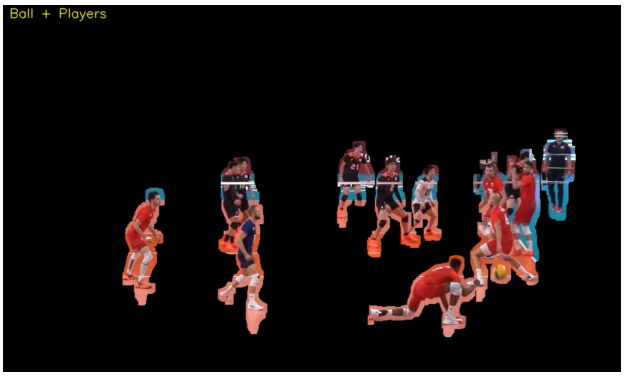
B. Limitations and Abnormal Observations

While the overall performance of the system is promising, several limitations were observed during experimentation. A key issue is the misclassification of semi-static objects, most notably the referee. Despite standing still for long periods, the referee is visually distinct from the learned background model and occasionally makes subtle movements. As a result, background subtraction algorithms such as MOG2 and KNN occasionally detect the referee as part of the foreground. This is problematic because contour filtering based solely on area cannot distinguish between a referee and an actual player, especially if their size is comparable. This leads to occasional

(a) Original Frame

(b) Players Only Frame

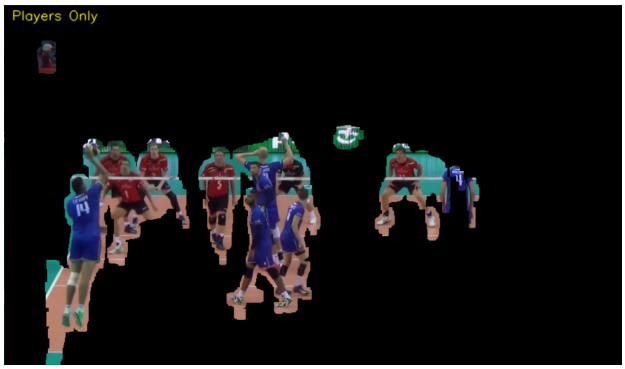
(c) Ball Only Frame



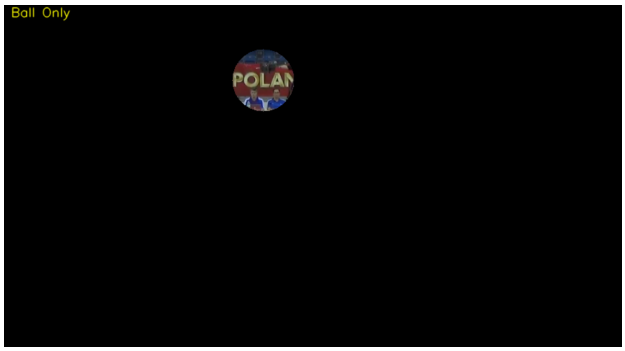
(d) Ball + Players Frame

Fig. 1. Failure case of the volleyball tracking pipeline: (a) Original input frame; (b) players detected using background subtraction; (c) false positives in ball detection caused by yellow ad banners; and (d) final output incorrectly including banner regions as the ball.

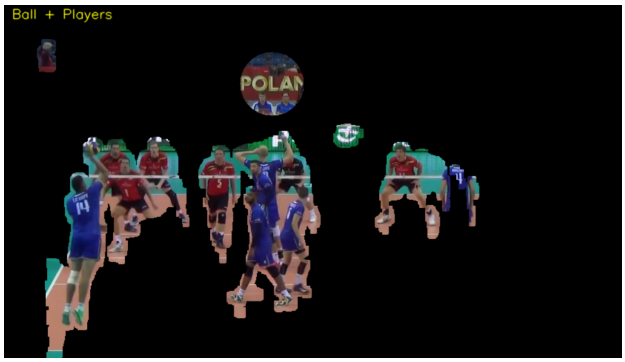
(a) Original Frame



(b) Players Only Frame



(c) Ball Only Frame



(d) Ball + Players Frame

Fig. 2. Failure case of the volleyball tracking pipeline: (a) Original input frame; (b) players detected using background subtraction; (c) false positives in ball detection caused by yellow ad banners; and (d) final output incorrectly including banner regions as the ball.

false positives in the "Players Only" and "Ball + Players" views, resulting in a less accurate representation of the actual game dynamics.

A significant challenge in ball detection arises due to the system's reliance on HSV color segmentation to identify yellow regions in the frame. This approach assumes consistent lighting and a clearly distinguishable ball color. However, when the volleyball is in motion, motion blur often reduces its contour size, making detection more difficult. In some instances, the system fails to detect the ball entirely. As illustrated in Figure 2, objects such as jersey highlights, banners, or reflections that fall within the predefined yellow range can be mistakenly identified as the ball, leading to false positives. Additionally, the ball's black stripe introduces further complications, as detection becomes unreliable when the stripe is facing the camera.

Moreover, while morphological operations like opening and closing help in cleaning up noise and connecting fragmented regions, they are highly sensitive to the chosen kernel size. For example, small kernels may fail to eliminate small blobs and shadow artifacts, while larger kernels may erode valid objects — especially the ball — from the masks entirely. There is no one-size-fits-all configuration, and achieving a good balance between noise reduction and object retention required manual tuning based on visual inspection of outputs.

The system also lacks temporal continuity or tracking logic. Each frame is processed independently without leveraging information from previous frames. This means the system cannot track player or ball movement over time or assign consistent object IDs across frames. Rapid or erratic motion (e.g., a spike or serve) may lead to temporal instability, with objects being inconsistently detected or disappearing briefly from the mask.

Additionally, if fans wear the same jersey as the players and move in the audience, they are not differentiated from the players. Along with this video advertisements are hard to detect as background.

C. Comparison to Existing Methods

When comparing our system to modern deep learning-based detection and tracking methods, the trade-offs between simplicity and robustness become apparent. Our method uses classical image processing techniques that do not require any training data, which is a major advantage in scenarios where annotated volleyball datasets are unavailable or difficult to produce. It runs in real-time on standard hardware and provides interpretable intermediate outputs (e.g., movement masks, ball masks), which are useful for debugging and visual verification.

However, deep learning methods — particularly object detectors like YOLOv5 or YOLOv8, instance segmentation models like Mask R-CNN, and pose estimators like OpenPose — offer significantly improved robustness, generalization, and multi-object tracking capabilities. These models can reliably detect and classify multiple players, distinguish between individuals, and accurately locate the ball even under occlusion, motion blur, or varying lighting conditions. Some are even trained on sports-specific datasets and can recognize actions such as serving or jumping.

Unlike our method, which is sensitive to manually defined color thresholds and contour areas, deep learning models learn complex spatial and appearance-based features that enable them to distinguish between players and non-players (such as referees or background spectators), or detect the ball based on shape, texture, and context rather than color alone. Furthermore, advanced systems often incorporate temporal models (e.g., optical flow, LSTMs, or multi-frame association) to provide stable tracking over time and support higher-level analytics such as player trajectories, ball speeds, and tactical positioning.

In summary, while our system provides a strong and lightweight baseline for volleyball tracking under controlled conditions, it lacks the adaptability and semantic understanding of modern machine learning approaches. Nonetheless, it serves as a valuable reference point and a practical tool for lightweight deployment, educational use, or as a preprocessing stage in more complex pipelines.

VI. Future Work

Deep learning models represent a modern and powerful approach to image analysis, enabling more precise and robust object detection compared to traditional methods. By leveraging **Deep Neural Networks (DNNs)**, we can classify different sections of a video frame and accurately determine the location of objects such as the ball and players. The implementation of deep learning techniques could significantly improve the accuracy and reliability of object tracking, even under challenging conditions.

A **Convolutional Neural Network (CNN)**, in particular, would be well-suited for this task, as it processes visual information in a hierarchical manner. CNNs extract low-level features such as edges, textures, and color gradients in the initial layers, while deeper layers identify more complex patterns, such as shapes and object structures. This ability to learn high-level representations allows CNN models to effectively distinguish between different objects, even in the presence of background noise, occlusions, or variations in illumination.

One of the primary challenges we encountered during our project was related to **the texture of the ball** and its visibility under certain conditions. During rapid rotations, variations in lighting, and the presence of shadows, the **HSV-based color segmentation** method struggled to consistently detect the ball. This inconsistency arises because the perceived hue and saturation of the yellow and blue stripes on the ball shift due to changes in illumination and motion blur, leading to **fragmented or inaccurate segmentation results**.

A deep learning-based model, such as a **CNN trained with a sufficiently diverse dataset**, could address this issue by learning to recognize the ball's texture, shape, and distinguishing features rather than relying solely on color-based heuristics. Such a model would be trained on a dataset containing

images of the ball under **various lighting conditions, viewing angles, partial occlusions, and rotational states** to ensure robust detection across different scenarios.

Moreover, integrating a **Recurrent Neural Network (RNN) or a Long Short-Term Memory (LSTM) network** alongside the CNN could enhance temporal consistency in tracking. These networks process sequential data and could help predict the ball's position in future frames based on its motion history, reducing the likelihood of false negatives in detection.

Another possible future enhancement is the implementation of **semantic segmentation networks**, such as **U-Net or DeepLabV3**, which would allow pixel-wise classification of objects in the scene. This would lead to more precise localization of the ball and players, reducing segmentation errors and improving tracking stability.

Due to time constraints, we were unable to fully explore and implement these advanced deep learning approaches with the level of accuracy required for real-world applications. Future work could involve **collecting a larger dataset**, fine-tuning **state-of-the-art CNN architectures such as ResNet, EfficientNet, or YOLO**, and exploring real-time implementations using **TensorFlow, PyTorch, or OpenVINO** for efficient inference on edge devices.

By incorporating deep learning-based approaches, we can significantly **improve the robustness, accuracy, and adaptability** of the system, enabling more effective and reliable object tracking in dynamic and complex environments.

ACKNOWLEDGMENT

We would like to express our gratitude to Prof. Shahata and the T.A.s for their invaluable guidance throughout the project. Additionally, we sincerely appreciate our colleagues, Ghaith and Hamza, for their support and assistance, even though they were part of other teams.

REFERENCES

[1] J. Bin, Z. Bahrami, C. A. Rahman, S. Du, S. Rogers, and Z. Liu, "Foreground fusion-based liquefied natural gas leak detection framework from surveillance thermal imaging," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 4, pp. 1151–1162, 2023.

[2] B. Garcia-Garcia, T. Bouwmans, and A. J. Rosales Silva, "Background subtraction in real applications: Challenges, current models and future directions," *Computer Science Review*, vol. 35, p. 100204, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013718303101

[3] P. Kurowski, K. Szelag, W. Zaluski, and R. Sitnik, "Accurate ball tracking in volleyball actions to support referees," *Opto-Electronics Review*, vol. 26, no. 4, pp. 296–306, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1230340218301045

[4] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 246–252, 1999.

[5] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," *International Conference on Pattern Recognition*, pp. 28–32, 2004.

[6] A. Salvador and J. R. Munkres, "Gaussian mixture models and their application to background subtraction," *Journal of Computer Vision*, vol. 89, pp. 1–16, 2009.

[7] W. Freeman and L. Gardner, "Dynamic gaussian mixture models for background subtraction," *IEEE Transactions on Image Processing*, vol. 10, no. 7, pp. 1036–1045, 2001.

[8] M. Hu, X. Yang, and Y. Chen, "Background subtraction using fuzzy gaussian mixture models," *Journal of Visual Communication and Image Representation*, vol. 26, pp. 238–245, 2015.

[9] J. Wang, Z. Li, and D. Zhang, "Foreground detection using texture and edge features for robust background subtraction," *Computer Vision and Image Understanding*, vol. 114, no. 6, pp. 703–710, 2010.

[10] S. Choudhury, P. Bhatnagar, and D. Jain, "Fuzzy gmms for background subtraction," *International Journal of Computer Vision*, vol. 12, pp. 22–32, 2008.

[11] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2, 2004, pp. 28–31 Vol.2.