VISION - M2 IMA

Practical work report
Object Tracking in Videos

Done by :

Carlos GRUSS & Oussama RCHAKI & Okba KHAREF

February 11, 2025

SORBONNE UNIVERSITÉ

Contents

1 – Objective

The goal of this practical work is to understand the challenges and difficulties of object tracking in videos, to experiment and develop solutions combining Mean Shift, Hough transform and Deep Features.

2 – Mean Shift

2.1 – Definition

The Mean Shift algorithm operates by iteratively refining a search window to find the region of maximum density in the feature space. In the context of object tracking, this feature space is often represented by a back-projection image, which serves as a similarity map. The back-projection is generated by computing the histogram of the Region of Interest (ROI) in the first frame and projecting it back onto subsequent frames to identify areas with similar color distributions.

The algorithm starts with an initial estimate of the object's position, defined by a search window. It then calculates the weighted centroid of the pixel intensities within this window, using the back-projection image as a guide. The search window is shifted iteratively to center around the computed centroid, progressively converging towards the densest region in the feature space. This iterative process continues until a predefined number of iterations is reached.

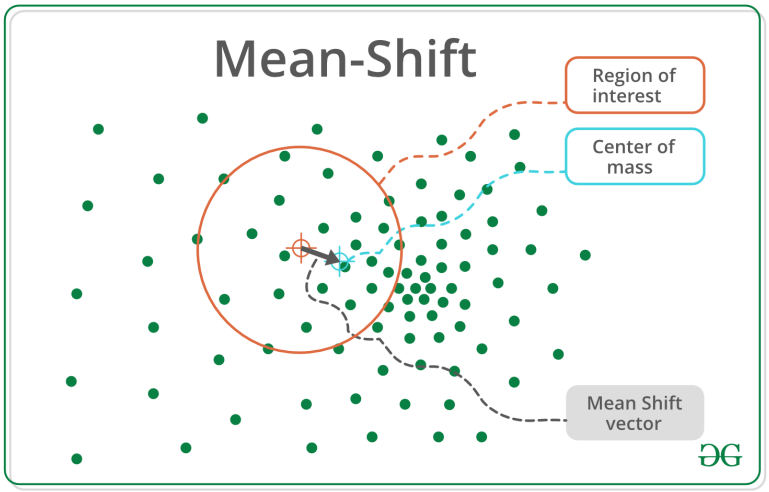
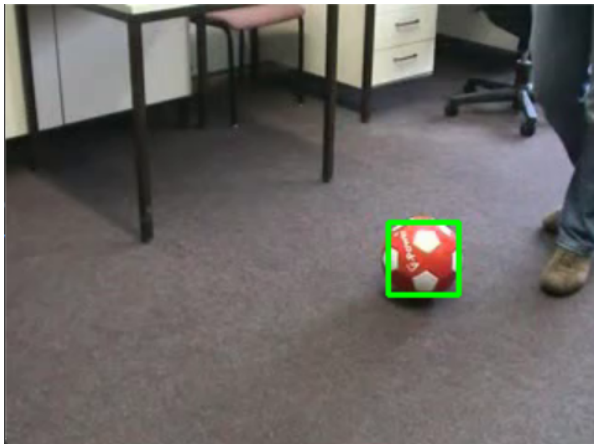
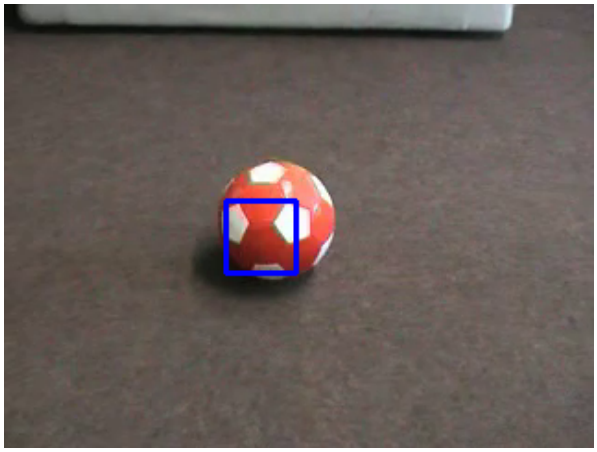


Figure 1 : Mean-Shift illustration

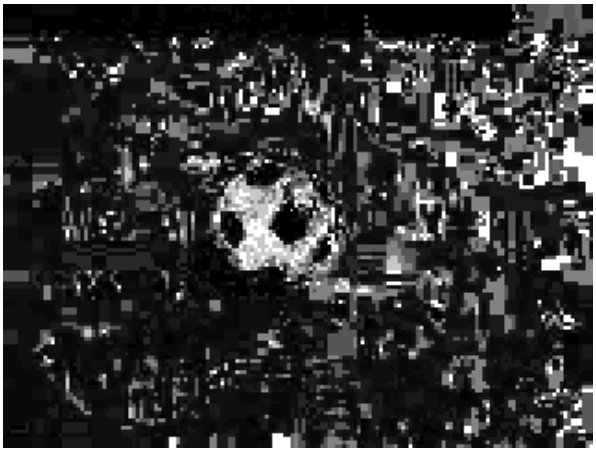
2.2 – First experience: advantages and limits



(a) Initialization.



(b) Search window in latter frame.

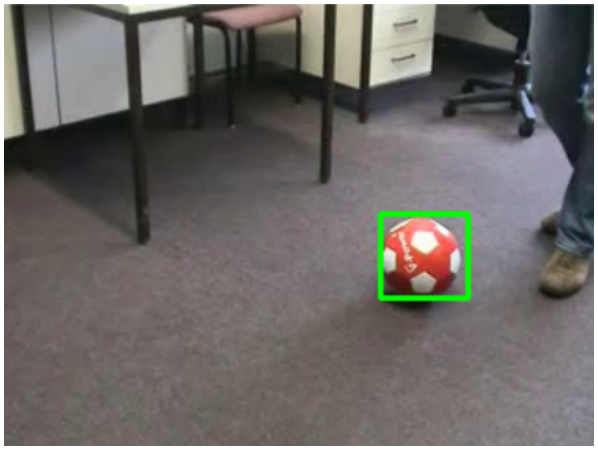


(c) Backprojection.

Figure 2 : Results of the first implementation.

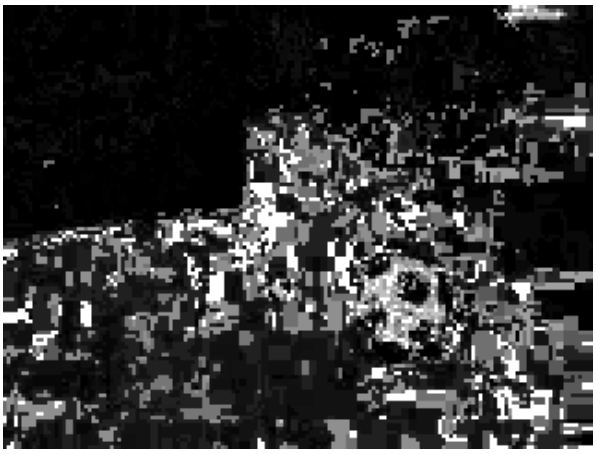
In the results of the first experiment, we observe that the method provides very effective tracking of the object (ball), especially in the initial frames. The method demonstrates robustness to outliers and performs well even under changes in illumination, thanks to the use of the hue component. This robustness can be also attributed to the back-projection technique, which relies on the object's color histogram rather than specific pixel values. By focusing on the distribution of color within the object, the method remains resilient to variations caused by noise, partial occlusions, or minor shape changes (such as when the ball moves slightly farther from the camera).

Another advantage of the method is its efficiency, as the Mean Shift algorithm performs a local search rather than a global search over the entire frame, making it computationally lightweight and suitable for real-time applications.



(A) Initialization.

(B) Search window in latter frame.



(C) Backprojection.

FIGURE 3 : Results of the first implementation with different initialization.

However, for other initializations, the tracking box fails to follow the object after a certain number of frames, especially when the object significantly changes its speed. Increasing the size of the search window was attempted, as it theoretically allows the algorithm to search a larger space to catch up with the object when its speed changes. However, this approach introduces new issues, as it increases the inclusion of outliers, which can cause the tracker to fail even in the very first frames. This indicates that the algorithm **heavily depends on proper initialization**.

Moreover, the algorithm is highly sensitive to noise. In scenarios where the video contains significant background noise or artifacts, the tracking accuracy can deteriorate quickly. This is because the algorithm relies heavily on the back-projection of the object's histogram, which can be easily disrupted by noise or similar colors in the background.

In addition, the algorithm struggles when the object disappears for a few frames (e.g., due to occlusion). In such cases, the tracker often loses the object entirely and fails to recover when it reappears. This limitation arises because the Mean Shift algorithm performs only a local search and does not consider temporal continuity, making it unable to reinitialize tracking after a temporary loss of visibility.

Furthermore, the fixed-size search window and lack of adaptability to scale changes make the algorithm unsuitable for scenarios where the object's size changes over time (e.g., the object moves closer or farther from the camera). This limitation reduces its robustness in real-world applications, where objects often undergo transformations in appearance and scale.

Finally, the use of the hue component helps make the algorithm robust to changes in illumination. However, in some scenes, the object can have a similar hue intensity as the background (Fig. 4). This highlights the Mean-Shift's strong dependency on initialization. Increasing the search window may include more background, further amplifying the risk of confusion between the object and the background

in later frames.

(A) Search window. (B) Hue component.

FIGURE 4 : Confusion of tracking the object due to the similarity between the object and the background in the hue component.

2.3 – Optimization

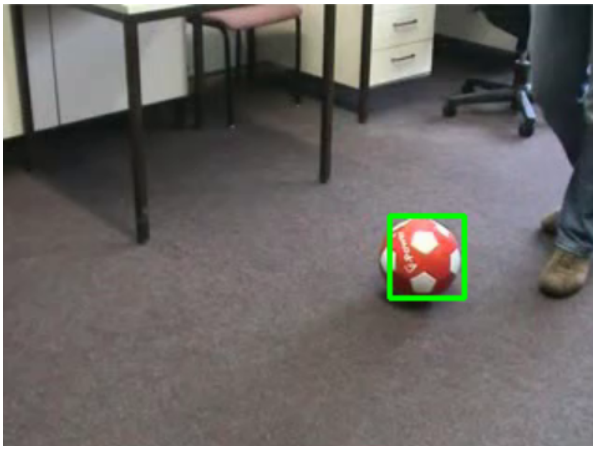
Updating the Model Histogram

In this section, we attempt to improve the tracking by updating the histogram of the tracked object dynamically during the tracking process. The current position of the object is extracted from `track_window`, and a new Region of Interest (ROI) is defined in the HSV color space using the updated bounding box coordinates.

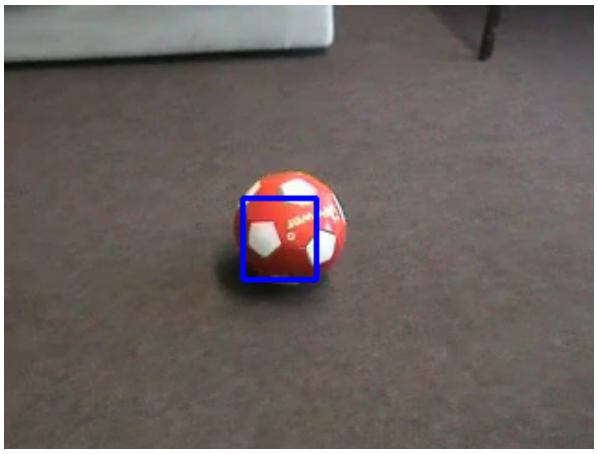
The histogram of the hue channel is recalculated for the updated ROI, normalized, and assigned to replace the previous histogram (`roi_hist`). This dynamic update allows the model to adapt to changes in the object's appearance over time, such as variations in illumination, subtle changes in color distribution, or changes in scale.

However, in practice, this solution did not perform as well as expected. If the bounding box fails to perfectly surround the object in a given frame, the updated histogram no longer accurately represents the true histogram of the object. This inaccuracy can progressively lead the algorithm to lose track of the object, reducing the overall robustness of the tracking system.

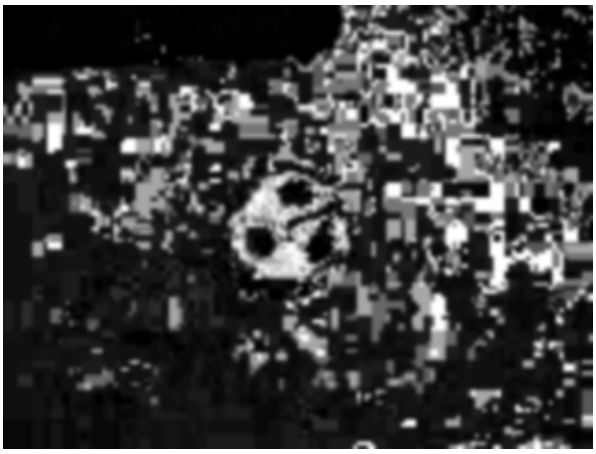
Smoothing



(A) Initialization.



(B) Search window in latter frame.



(C) Backprojection.

Figure 5 : Results after smoothing the Backprojection with a gaussian filter.

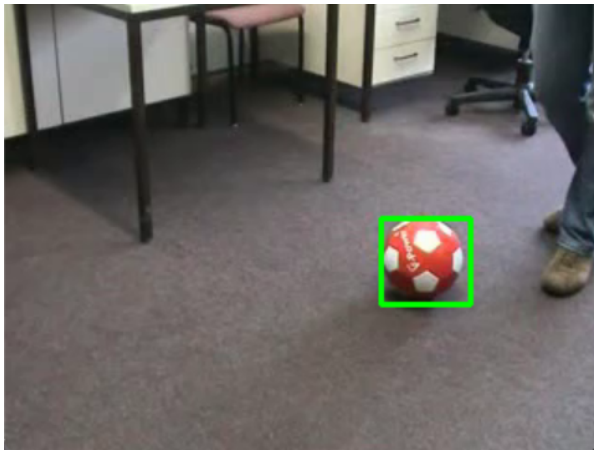
In this part, we optimize the algorithm by applying a Gaussian filter to smooth the back-projection image. The back-projection, which represents the likelihood map of the object's location based on its color histogram, can often contain noise or abrupt intensity changes. These artifacts may arise from background elements, lighting variations, or imperfections in the histogram calculation.

To address this issue, a Gaussian filter of size (5,5) is applied to the back-projection image. This smoothing process helps reduce high-frequency noise and creates a more uniform likelihood map, making the algorithm more robust. By eliminating abrupt intensity fluctuations, the Mean Shift algorithm can converge more reliably toward the object's actual location, avoiding false positives or misalignments caused by noise.

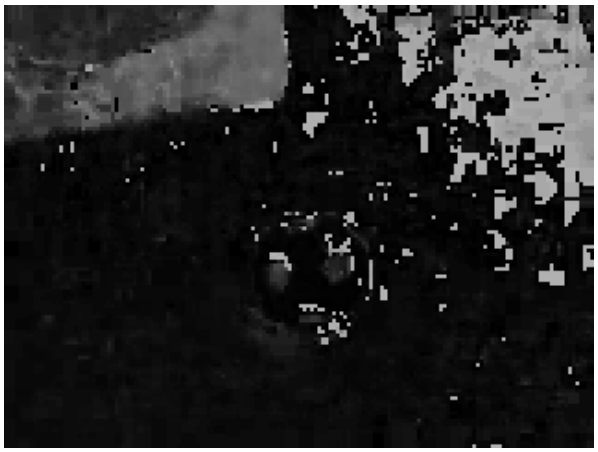
In the video used here, there was no significant noise present. However, in scenarios where noise is significant, it would be beneficial to smooth the frames beforehand to reduce noise. This preprocessing step would help ensure a cleaner representation of the object's histogram, improving the accuracy and robustness of the tracking algorithm.

This optimization yielded good results; however, we still observed that, for some (rare) initializations, the algorithm consistently failed to track the ball. Therefore, we propose another method to achieve more reliable tracking results, making the model less dependent on the initialization.

Using the hue and saturation compenents

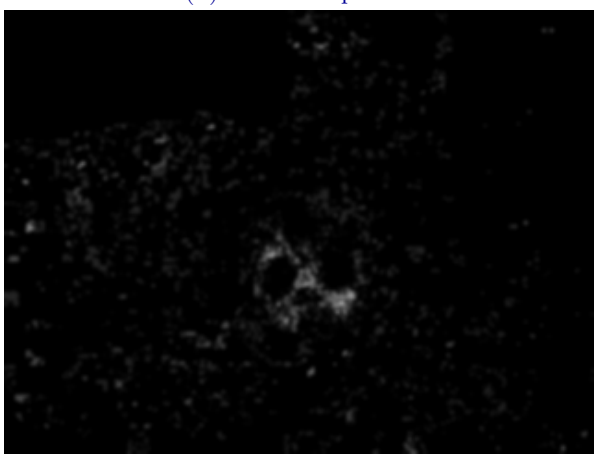


(A) Initialization.

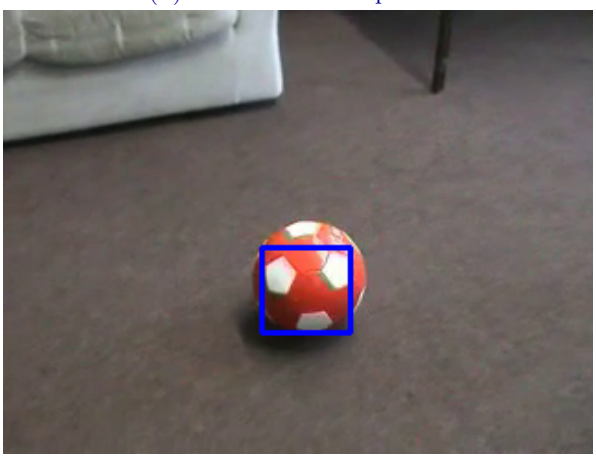


(B) Hue component.

(C) Saturation component.



(D) Backprojection.



(E) Search window.

Figure 6 : Results after taking into account the hue and saturation components.

In this section, instead of using only the Hue component to calculate the histogram, we use both Hue and Saturation. This decision was based on our observation that, in frames where the object has the same intensity as the background in the Hue component, the object often has distinguishable intensity in the Saturation component. The idea was to combine the independence from luminosity provided by the Hue component with the distinguishable representation of the object in the Saturation component. This combination ensures robust tracking of the object across all frames and for all (acceptable) initializations.

And indeed, the results confirmed this approach was highly effective.

3 – Hough Transform

3.1 – Definition

The Generalized Hough Transform (GHT) is an extension of the classical Hough Transform, designed to detect arbitrary shapes that lack a simple parametric equation, such as circles or ellipses. Unlike the standard method, which relies on predefined equations, the GHT uses a model-based approach to recognize objects based on their shape characteristics.

The GHT is based on the concept of a reference model and an R-table. Instead of directly parameterizing lines or curves, it utilizes shape descriptors extracted from the contour of an object.

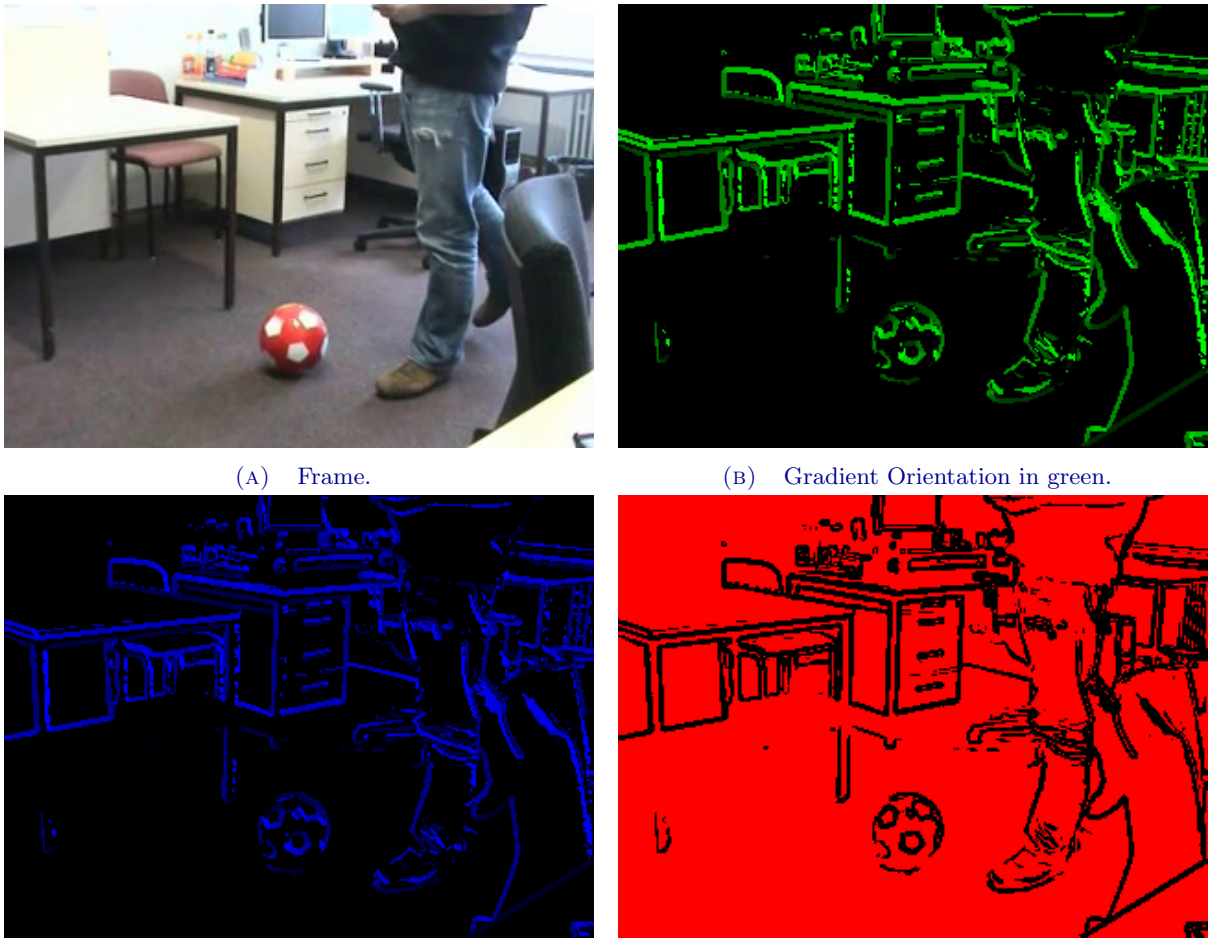
To apply the GHT, a reference shape is first defined, typically extracted from an example object. The steps involved in constructing the model are:

1. Identify edge points of the reference shape.
2. Compute a descriptor for each edge point, such as gradient direction or curvature.
3. Define a reference point (e.g., the centroid of the shape).
4. Store the relative position of each edge point with respect to the reference point in an R-table.

The R-table acts as a look-up table where each gradient direction is associated with a set of displacement vectors pointing to possible reference point locations.

Once the model is constructed, detection in a target image follows these steps:

1. Apply an edge detection algorithm (e.g., Canny) to extract the contours.
2. Compute gradient orientation for each edge point in the image.
3. Use the R-table to vote for possible reference point locations in an accumulator space.
4. Identify peaks in the accumulator space, which correspond to probable locations of the target shape.



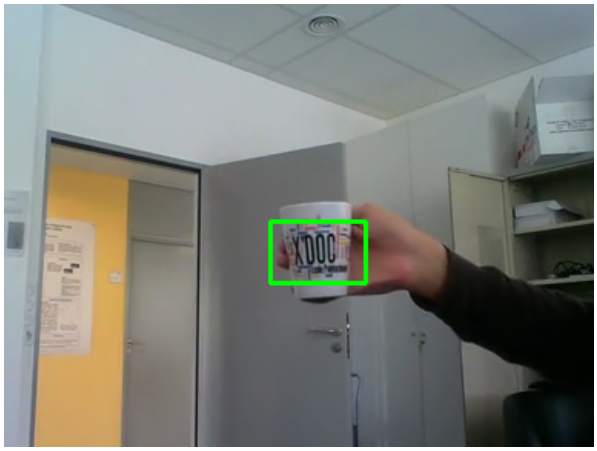
(A) Frame. (B) Gradient Orientation in green.

(C) Gradient magnitude in blue. (D) Masked pixels in red.

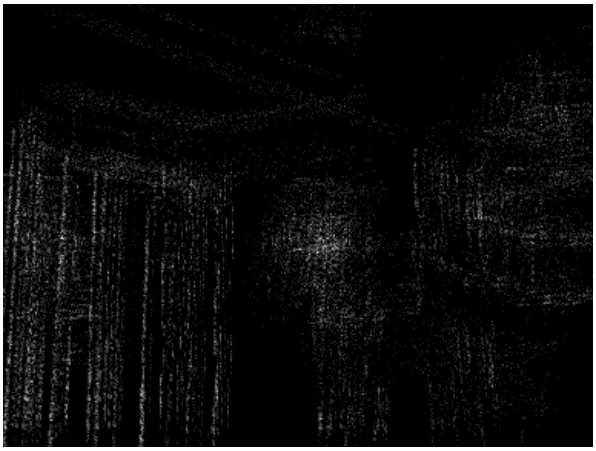
FIGURE 7 : Computing the index of the vote (gradient orientation), and selection of the voting pixels (using the gradient norm). .

3.2 – Implementation and optimization

(A) Object to track.



(B) Object tracking.



(C) Accumulator.

(D) Gradient orientation.

(E) Gradient magnitude.

Figure 8 : Results of our implementation.

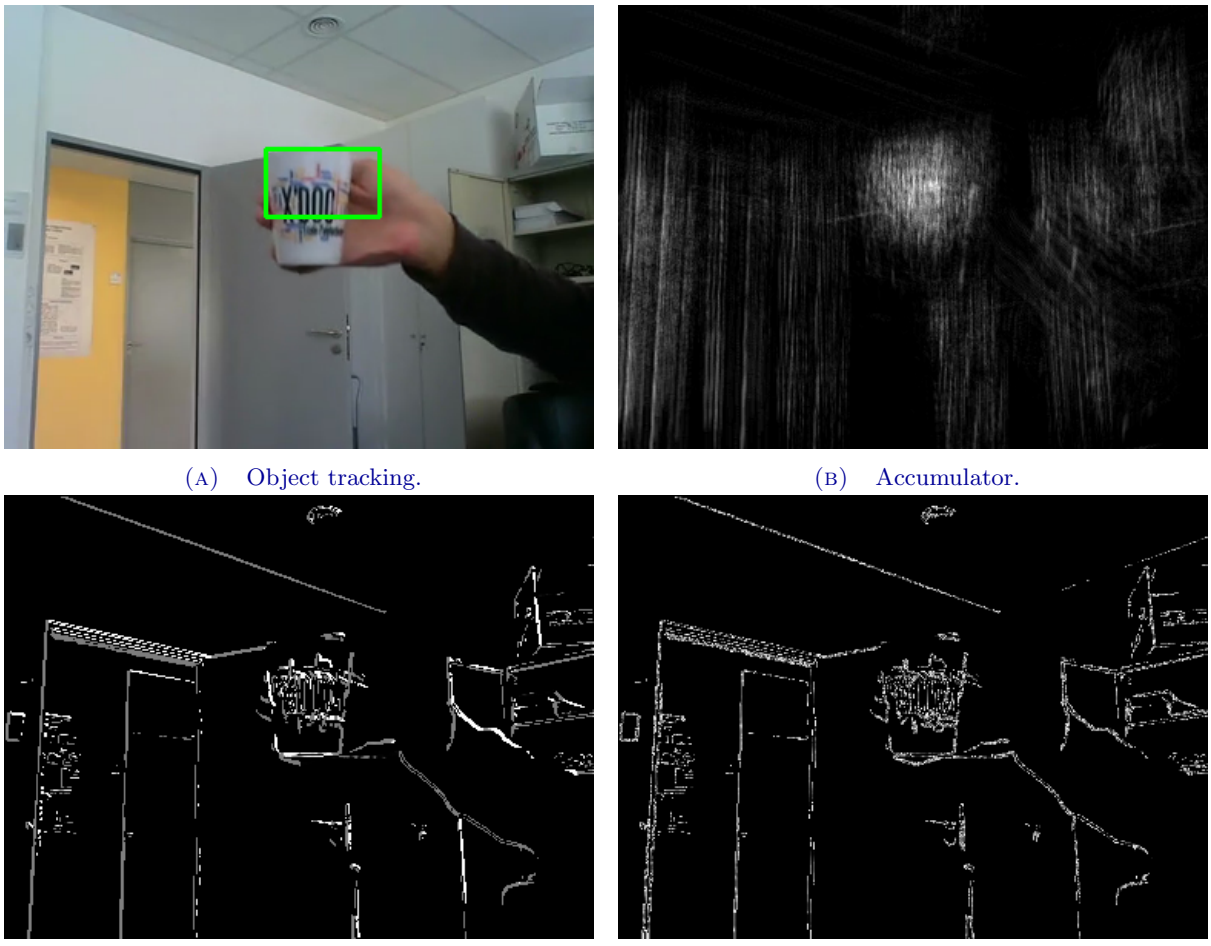
We successfully implemented the Generalized Hough Transform and achieved effective tracking of the mug. This method demonstrates the ability to detect arbitrary shapes beyond simple parametric curves and exhibits robustness to partial occlusions. Even if the object disappears for certain frames, it can be reliably re-detected afterward, unlike the Mean-Shift method. Additionally, the Generalized Hough Transform is resilient to noise and capable of handling varying object velocities, provided that the object's

image remains clear, this tracking method ensures consistent detection.

However, in frames where the object moves rapidly, motion blur occurs (flou de bougé), causing the object's image to become blurry and resulting in the loss of edge information. As a consequence, the object is no longer detected. Nevertheless, once the object appears clearly again, the Generalized Hough Transform (GHT) can successfully re-detect it. This characteristic can also be seen as an advantage, as even if tracking is temporarily lost, it can be regained once a sharp image of the object is available.

Additionally, the disadvantages of GHT include high computational cost, making it less suitable for real-time applications, high memory consumption, due to the large accumulator space required, and difficulty in distinguishing closely spaced shapes, as overlapping votes in the parameter space can lead to ambiguity in detection.

To optimize this method, we observed that the mug contains text with letters featuring prominent vertical edges. To enhance detection, we prioritized the votes of points belonging to vertical edges by multiplying $Grad_x$ by 100. This adjustment significantly improved the results, as shown in the figure below. However, despite this improvement, the object is still lost in frames where motion blur occurs.



(A) Object tracking. (B) Accumulator.

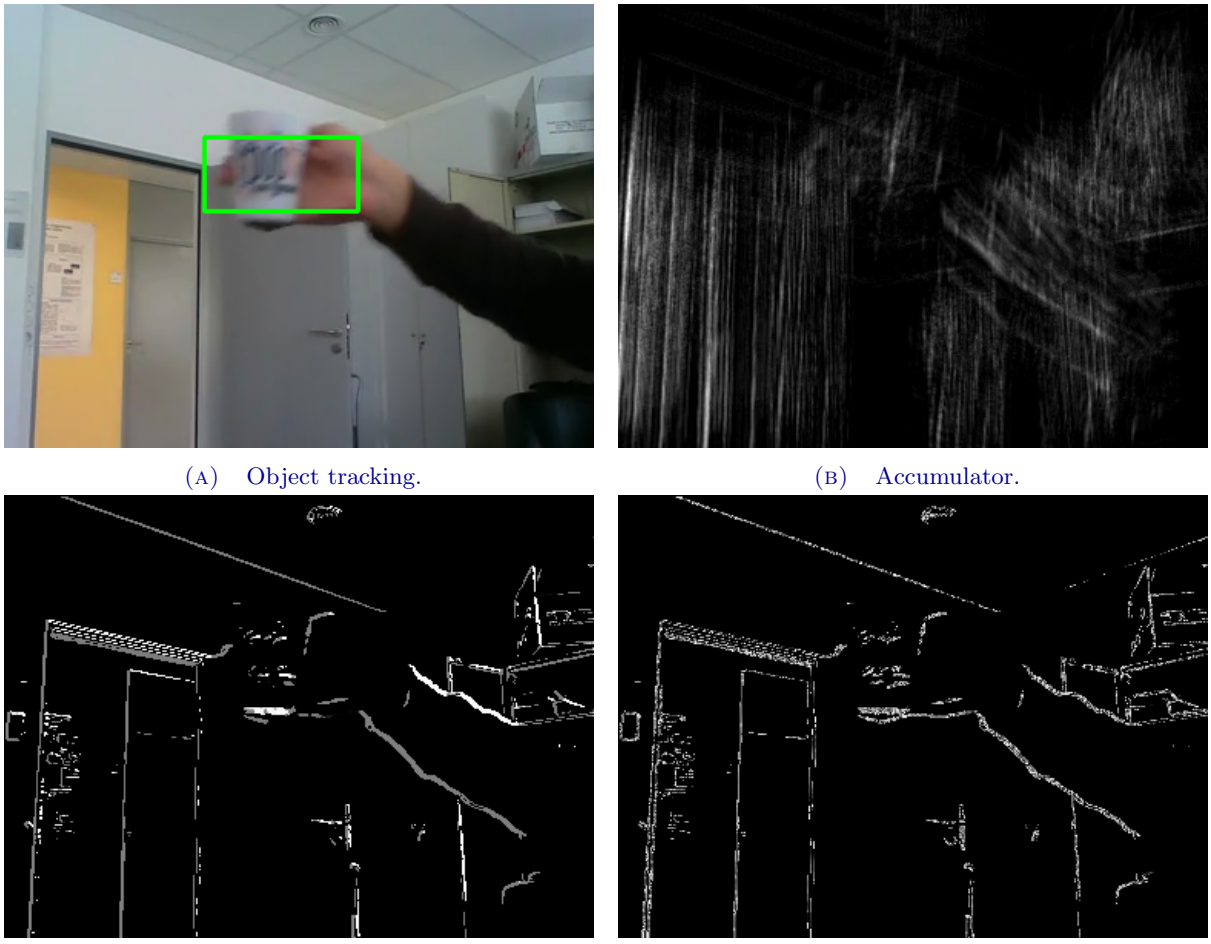
(C) Gradient orientation. (D) Gradient magnitude.

FIGURE 9 : Results of our implementation with the first optimization.

3.3 – Exploiting the smoothness of the displacement

In this part, we enhance the Generalized Hough Transform (GHT) by leveraging the smooth displacement of the mug for more stable tracking. We introduce motion prediction by estimating the next position based on the object's previous velocity, allowing tracking to continue even when detections are momentarily lost due to blur or occlusions. To prevent false detections, we implement a threshold-based filtering that ignores unrealistic jumps in position, ensuring smooth tracking. Additionally, we enhance

edge detection by amplifying horizontal gradients, which strengthens vertical features, particularly useful for tracking text on the mug. Finally, we ensure that velocity updates only occur with valid detections, preventing unwanted resets and maintaining motion continuity. These modifications improve robustness, reduce tracking errors, and make the method more reliable in dynamic conditions.



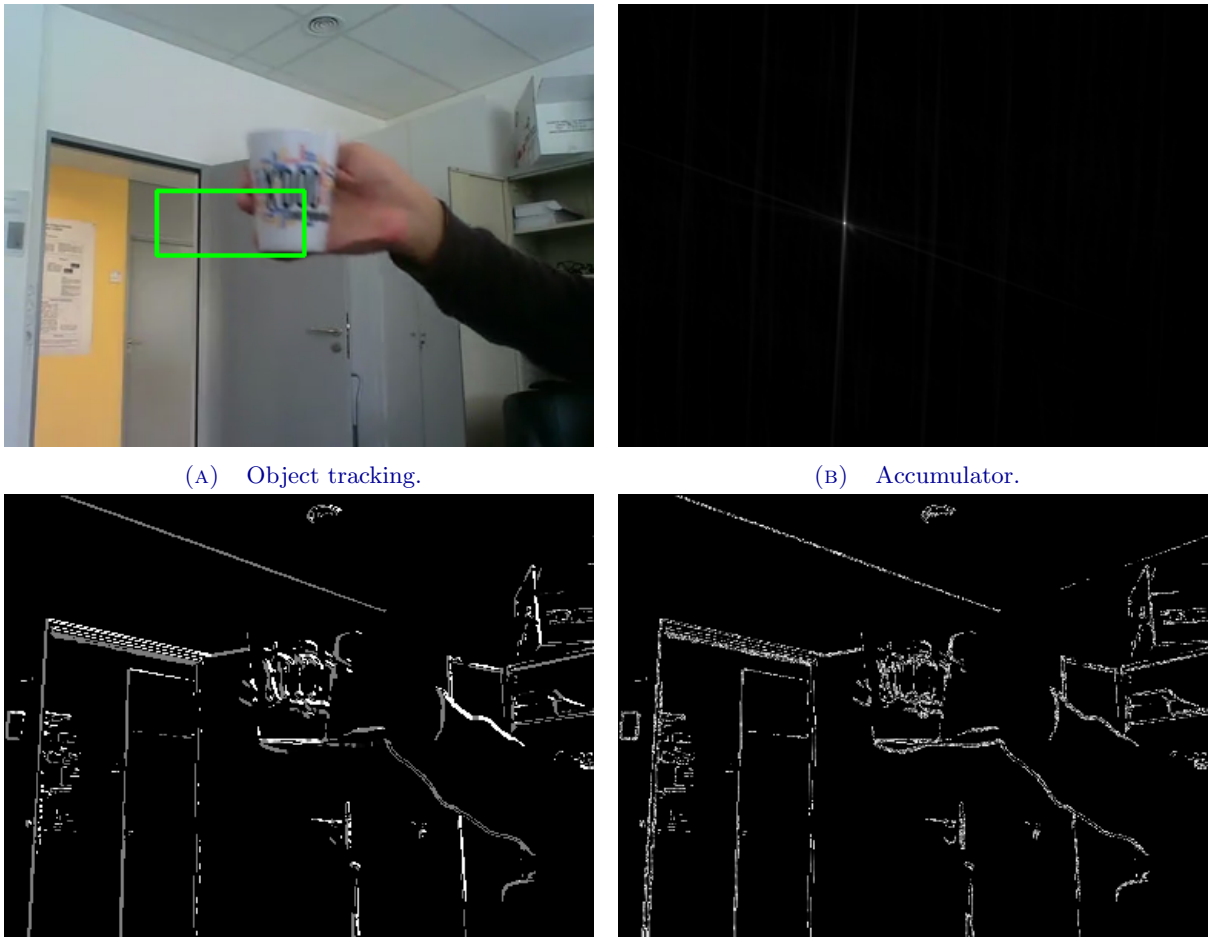
(A) Object tracking. (B) Accumulator.

(C) Gradient orientation. (D) Gradient magnitude.

FIGURE 10 : Results of our implementation by exploiting the smoothness of the displacement.

Our method implemented showed nothing ... BUT outstanding results, as the figure above show. Although the accumulator does not display a maximum vote at the mug's position due to motion blur caused by its displacement (as the mug's gradient orientation and magnitude are no longer visible), our approach successfully tracks the mug by predicting its position based on its displacement velocity. This prediction ensures continuous tracking until the Generalized Hough Transform (GHT) identifies the mug again, once it reappears clearly, allowing us to seamlessly resume tracking using GHT.

3.4 – Update strategy



(A) Object tracking. (B) Accumulator.

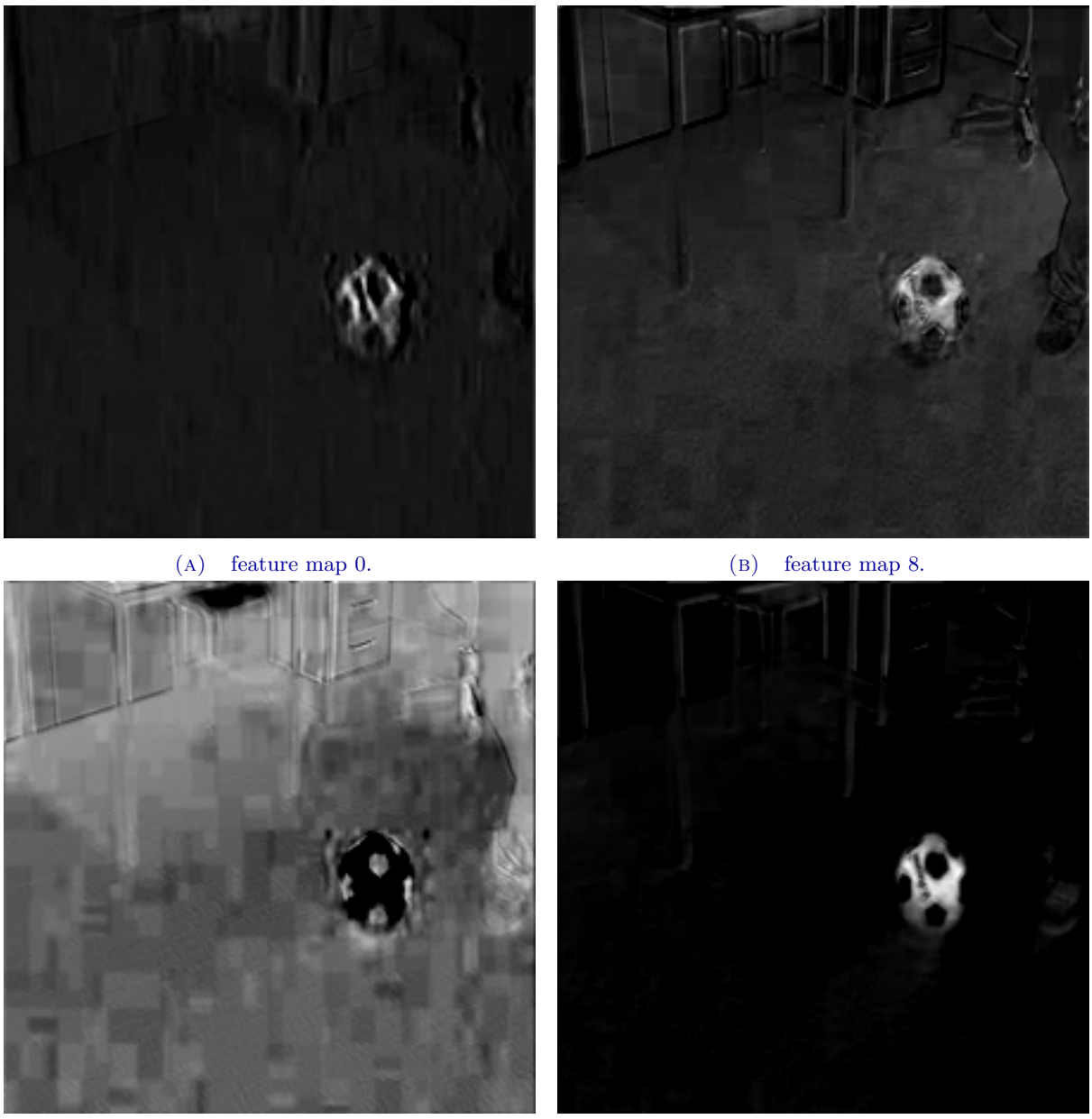
(C) Gradient orientation. (D) Gradient magnitude.

Figure 11 : Results of our implementation by updating the reference point and the r-table.

In this section, we implement a simple strategy to update the model, making it more robust to aspect changes and occlusions. Specifically, we update the reference point and the R-table for each frame to adapt to variations in the object's appearance and improve tracking stability.

However, this method is computationally intensive. While it successfully maintains tracking without losing the object, it struggles to perfectly enclose and sharply detect it.

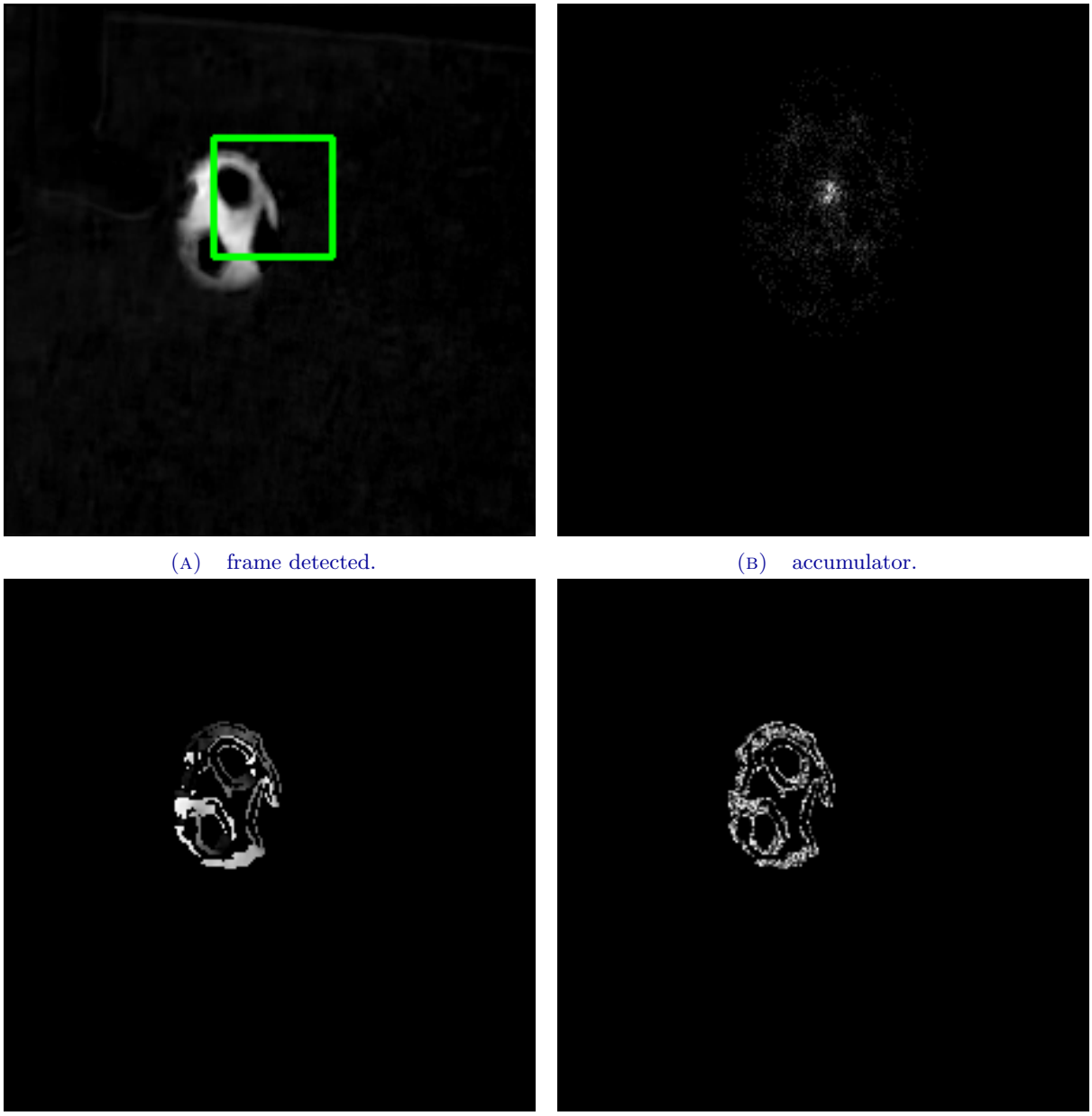
4 – Deep features



(A) feature map 0. (B) feature map 8.

(C) feature map 48. (D) feature map 61.

Figure 12 : Map feature corresponding the output of the first convolutional layer + its activation function of VGG16.



(A) frame detected. (B) accumulator.

(C) orientation masked. (D) magnitude.

FIGURE 13 : Result of using the 61th Map feature with GHT.

We used VGG16 trained on ImageNet and extracted the output of the first convolutional layer along with its activation function. We chose the first layer because it preserves a high-resolution representation of the image, the first layer produce 64 feature maps, we selected the 61st channel as it is the one that best distinguishes the ball from other objects.

The results were not highly satisfying, as we lost a lot of detail in the object (the ball), making the gradients less relevant. However, throughout the video, the tracking of the ball was not lost.

However, it was unfortunate that we couldn't go further in this part due to time constraints. Nevertheless, significant improvements can be made by identifying feature maps that best represent vertical edges and those that best capture horizontal edges. These two selected feature maps could then be used in the GHT algorithm instead of $Grad_x$ and $Grad_y$.