Project Documentation: Sleep Motion Tracker (Proof of Concept)

1. Executive Summary

The Sleep Motion Tracker (PoC) was developed as an experimental prototype to explore the use of computer vision for analyzing body movement during sleep.
The project demonstrates the ability to design a robust motion-detection pipeline capable of operating under low-light conditions using infrared (IR) imaging.

While initially conceived as a simple prototype, the system establishes a foundation for future expansion into a more comprehensive sleep analysis platform — potentially integrating wearable sensors, environmental data, and machine learning models for behavioral analysis.

2. Project Description

The goal of the project is to create a system for the objective recording and quantification of sleep movements, serving as a preliminary building block for a modular sleep-lab system.

Hardware: Raspberry Pi 4, CSI infrared night-vision camera
Software: Python, OpenCV (Computer Vision Library)
Methodology: Adaptive background subtraction using MOG2
Output: Logs timestamp, motion state (yes/no), and motion area to a CSV file

The system is designed to continuously monitor a sleeping subject using an IR camera and to detect significant body movements through real-time image analysis.
All detected motion events are timestamped and logged for later data visualization and analysis.

3. Technical Implementation (OpenCV Approach)

The main challenge in this project is to reliably distinguish a moving object (the sleeper) from a static background (the bed) under minimal lighting.

1. Camera Initialization

The application uses the cv2.VideoCapture interface, which can be adapted for the Raspberry Pi’s CSI camera module (e.g., through picamera2 integration).

2. Background Modeling

The algorithm cv2.createBackgroundSubtractorMOG2 is employed to dynamically model the background.
This adaptive approach allows the system to handle gradual changes in illumination and avoids the limitations of static background subtraction methods (like Absolute Difference), which would quickly fail in realistic environments.

3. Noise Reduction

Infrared video streams tend to be noisy, particularly in low-light settings.
To address this:
- Gaussian Blurring (cv2.GaussianBlur) smooths image noise and reduces false positives.
- Morphological Operations (Erosion and Dilation, using cv2.morphologyEx) remove small irrelevant pixel clusters and refine motion contours.

4. Motion Detection and Filtering

- Contour Detection: Using cv2.findContours, the system identifies motion regions within the processed frame.
- Area Thresholding: Only contours with an area above a predefined constant (MIN_CONTOUR_AREA) are classified as valid movements, preventing minor pixel fluctuations (such as dust or IR flicker) from being logged.

5. Data Logging

Detected movement events are stored in a CSV file, containing:
- Timestamp
- Motion state (True/False)
- Maximum contour area

This enables subsequent statistical or graphical evaluation of motion frequency and intensity throughout the night.

4. Conclusion and Outlook

The Sleep Motion Tracker demonstrates how fundamental image-processing techniques can be effectively applied to motion analysis in real-world conditions.
It combines principles from computer vision, signal filtering, and data logging in a lightweight Python-based implementation.

Key Insights:
- Adaptive background modeling (MOG2) significantly improves robustness under fluctuating light conditions.
- Morphological filtering is essential for reducing false positives from sensor noise.
- Even low-resolution IR video data is sufficient for accurate detection of macroscopic motion.

Future Development
The project is designed to be modular, allowing easy integration with additional systems and sensors:
- Integration of wearable modules (IMU, pulse oximeter) for physiological context
- Synchronization and data upload via BLE or MQTT
- Visualization of motion timelines in Grafana or similar dashboards
- Edge ML models for posture classification and sleep-phase inference
- Multi-sensor fusion combining camera and acoustic data

5. Files and Resources

- sleep_motion_tracker.py – Main Python script implementing motion tracking and logging
- motion_log.csv – Example output file after a short test session
- Documentation (this file)