**1. Introduction and Related Works**

Understanding honey bee activity is crucial for monitoring hive health and ensuring optimal environmental conditions for colony survival. Traditional beekeeping methods rely heavily on manual inspections, which are time-consuming and intrusive. With advancements in computer vision, it is now possible to monitor hives unobtrusively and automatically by analyzing visual data. This project explores the use of classical computer vision techniques for detecting, tracking, and analyzing honey bee movement at the hive entrance using a top-down camera setup.

The primary objective of this work is to develop a pipeline that uses classical techniques for robust background subtraction, bee detection, orientation estimation, and persistent ID tracking. The intended application is to assist beekeepers by providing real-time data on bee ingress and egress behavior without relying on computationally expensive neural network models.

The inspiration for this work comes from a variety of existing studies that utilize different computer vision methodologies for similar applications. Among the notable contributions, Lowe [1] introduced the Scale-Invariant Feature Transform (SIFT), which has proven robust for feature detection in insect imagery. Sledevič et al. [2] built on this by using keypoint-based methods for bee orientation detection, though their methods rely on higher quality images. Similarly, Dalal and Triggs [3] proposed Histogram of Oriented Gradients (HOG) to detect object orientations, though HOG is more tuned to object detection rather than body orientation.

Contour and shape-based methods also offer insight into bee pose estimation. Active contours ("snakes") as proposed by Kass et al. [4], and ellipse fitting by Fitzgibbon et al. [5], help delineate object boundaries and approximate orientation. However, these methods may falter in low-contrast or noisy environments. Lindeberg [6] and Grigorescu et al. [7] addressed such challenges by using blob and texture detection methods, though computational efficiency remains a concern.

Motion-based analysis, particularly optical flow techniques introduced by Horn and Schunck [8], enables orientation and velocity estimation by analyzing frame-to-frame pixel shifts. Lingenfelter et al. [9] explored insect-inspired motion estimation using optic flow, reinforcing the applicability of such methods in bee tracking. While learning-based models, such as those by Robinson et al. [10] and Sanaullah et al. [11], show promise in adapting to complex environments, they exceed the intended scope of this project, which favors classical techniques.

Template-based approaches (Brunelli [12]) and Hough Transforms (Duda et al. [13]) offer additional classical tools for shape detection. Behavioral studies such as those by Blut et al. [14] and datasets provided by Chaudhary et al. [15] support the development of bee behavior tracking systems. Though Cheng et al. [16] and Stürzl and Möller [17] focus on other insects or panoramic imaging, their methods contribute conceptual foundations for insect orientation analysis.

Together, these studies provide a foundation for building a real-time, classical computer vision-based honey bee monitoring system. The unique contribution of this project lies in integrating these classical techniques into a cohesive tool for real-time bee detection, orientation estimation, and velocity visualization.

**2. Project Design or Architecture**

The architecture of the proposed system consists of five core components: (1) video acquisition, (2) background subtraction, (3) contour extraction and feature estimation, (4) bee tracking with persistent IDs, and (5) orientation and velocity visualization.

Video is captured from a top-down camera mounted above the hive entrance. The video stream is first converted to grayscale and smoothed using Gaussian blur. A background model is created using a running average method (cv2.accumulateWeighted) to dynamically adapt to changing lighting conditions and environmental noise.

Foreground segmentation is performed by subtracting the background model from the current frame and thresholding the result. Morphological operations remove noise. Contours are then extracted and filtered based on area to isolate bees from small debris or shadows.

Centroids of the filtered contours are passed to a BeeTracker class, which uses a distance-based matching algorithm to maintain persistent IDs. Each bee’s trajectory is stored for visualization and analysis. Orientation is estimated using PCA on each bee’s contour, and velocity is approximated using optical flow vectors. All outputs are rendered as overlays on the video feed in real time.

**3. Implementation and Evaluation of the Proposed Approach**

The implementation is done entirely in Python using OpenCV and NumPy. A sample video (bees.mp4) is used as input. The background subtraction method employs a weighted running average, which balances adaptiveness with robustness against transient movement such as wind-blown grass.

Contours are extracted using cv2.findContours, and bees are filtered based on contour area. Centroids are computed via image moments, and stored in a dictionary with unique IDs. If a new centroid is close to an existing one from the previous frame, it is assigned the same ID.

To visualize trajectories, each bee’s position over time is stored and drawn as a polyline. The PCA-based orientation estimation uses the major eigenvector of the contour's point cloud to determine the bee’s heading. Velocity vectors are estimated using the Farneback method for dense optical flow, and directional arrows are drawn for visual feedback.

The evaluation was qualitative. The system correctly isolated and tracked individual bees, providing consistent IDs across frames. Trajectories and velocity vectors aligned with observable movement, and orientation angles were stable. Performance remained real-time (~30 fps) on consumer hardware. The modularity of the code supports easy parameter tuning (e.g., minimum contour size, flow threshold) for adapting to other lighting conditions or hive configurations.

Although the approach is effective, it occasionally suffers from ID switching during occlusions or rapid bee movement. Small debris can also be misclassified as bees if not filtered carefully. These issues may be mitigated by combining this classical approach with lightweight learning-based refinements in future iterations.

**4. Conclusion and Discussion**

This project demonstrates the effectiveness of classical computer vision techniques for honey bee tracking at the hive entrance. The integration of background subtraction, contour-based detection, PCA orientation estimation, and optical flow velocity estimation creates a robust, real-time monitoring system.

While the system is limited in dealing with heavy occlusions or overlapping bees, it performs well under typical field conditions and requires minimal computational resources. This makes it suitable for deployment on embedded systems or edge devices. By avoiding reliance on neural networks, the solution remains interpretable, tunable, and lightweight.

Future improvements could include incorporating lightweight tracking filters (e.g., Kalman filters), integrating environmental context (e.g., weather data), or expanding to 3D orientation models. Additionally, exporting trajectory data for offline analysis would further enhance its utility for scientific or beekeeping purposes.

**5. References**

[1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

[2] T. Sledevič et al., “Keypoint-Based Bee Orientation Estimation and Ramp Detection at the Hive Entrance for Bee Behavior Identification System,” Agriculture, vol. 14, no. 11, p. 1890, 2024.

[3] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[4] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active Contour Models,” International Journal of Computer Vision, vol. 1, no. 4, pp. 321–331, 1988.

[5] A. Fitzgibbon et al., “Direct Least Square Fitting of Ellipses,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 5, pp. 476–480, 1999.

[6] T. Lindeberg, “Detecting Salient Blob-like Image Structures,” International Journal of Computer Vision, vol. 11, no. 3, pp. 283–318, 1993.

[7] S. E. Grigorescu et al., “Comparison of Texture Features Based on Gabor Filters,” IEEE Transactions on Image Processing, vol. 11, no. 10, pp. 1160–1167, 2002.

[8] B. K. P. Horn and B. G. Schunck, “Determining Optical Flow,” Artificial Intelligence, vol. 17, no. 1–3, pp. 185–203, 1981.

[9] B. F. Lingenfelter et al., “Insect Inspired Vision-Based Velocity Estimation,” Bioinspiration & Biomimetics, 2023.

[10] B. S. Robinson et al., “Online Learning for Orientation Estimation During Translation in an Insect Ring Attractor Network,” Scientific Reports, vol. 12, p. 5798, 2022.

[11] Sanaullah et al., “Exploring Spiking Neural Networks,” Frontiers in Computational Neuroscience, 2023.

[12] R. Brunelli, Template Matching Techniques in Computer Vision: Theory and Practice. Wiley, 2009.

[13] R. O. Duda et al., “Use of the Hough Transformation to Detect Lines and Curves,” Communications of the ACM, vol. 15, no. 1, pp. 11–15, 1972.

[14] C. Blut et al., “Automated Detection of Encounter Behaviors,” Scientific Reports, 2017.

[15] P. Chaudhary et al., “Honey Bee Image Dataset,” Mississippi State University Libraries, 2024.

[16] Y. Cheng et al., “Estimating Orientation of Flying Fruit Flies,” PLoS One, 2015.

[17] W. Stürzl and R. Möller, “Insect-Inspired Active Vision Approach,” SpringerLink, 1970.