Conveyor Belt Speed Estimation

Vinat Goyal

December 2024

1 Introduction

Speed estimation of objects on conveyor belts is a critical task in industrial automation, enabling real-time monitoring and quality assurance in manufacturing. Traditional methods rely on hardware-based solutions such as tachometers, which can be expensive and challenging to integrate with existing systems.

This report explores a computer vision-based approach that leverages feature detection and tracking to estimate conveyor belt speeds from the camera data. This approach is cost-effective, scalable, and adaptable to different operational environments. The focus is on using the SIFT (Scale-Invariant Feature Transform) algorithm for feature detection, matching, and Kalman filters for smoothing speed measurements.

2 Literature Review

Speed estimation of moving objects in videos is a critical task in industrial automation. Traditional hardware-based solutions come with high costs, limited flexibility, and dependency on specific setups. Computer vision-based methods offer a scalable and cost-effective alternative, leveraging advancements in feature detection, tracking, and modern machine learning models[2].

2.1 Preprocessing: Focusing on Relevant Regions

Preprocessing plays a crucial role in any computer vision task in reducing computational overhead and improving the accuracy of feature detection and matching. In real-world environments, conveyor belts often operate in cluttered backgrounds. Gaussian Mixture Models (GMM)[8] are widely used for background subtraction in such scenarios, effectively separating moving objects from the static background.

Another preprocessing step involves masking the region of interest using polygonal boundaries. By limiting processing to the belt region, noise from irrelevant areas is minimized, enhancing both the speed and reliability of the pipeline.

2.2 Feature Detection

Feature detection plays a fundamental role in identifying unique points in consecutive video frames, which is the basis for object motion tracking. Early research by David Lowe introduced the Scale-Invariant Feature Transform (SIFT)[4], which extracts distinctive key points that are invariant to scaling, rotation, and partial occlusion.

Later, ORB (Oriented FAST and Rotated BRIEF)[6] was developed to address the computational inefficiency of SIFT. ORB combines a fast keypoint detection algorithm with efficient binary descriptors, making it suitable for real-time applications. Similarly, SURF (Speeded-Up Robust Features)[1] provides a faster alternative to SIFT by simplifying the descriptor calculation, making it ideal for low-resource environments.

In the context of conveyor belt speed estimation, these feature detectors identify regions of interest in moving objects (e.g., edges, corners) that can be tracked across frames. However, the choice of detector depends on the specific application: SIFT excels in precision, while ORB and SURF offer faster processing suitable for real-time industrial settings.

2.3 Feature Matching

Once features are detected, establishing correspondences between features in consecutive frames is critical for motion estimation.

The simplest and most intuitive approach to feature matching is Brute Force Matching. In this method, descriptors from one frame are compared exhaustively with descriptors from the next frame to find the closest match. While brute force matching guarantees finding the best matches, it is computationally expensive, especially for large datasets or high-dimensional descriptors like SIFT or SURF.

Matching algorithms such as the Fast Library for Approximate Nearest Neighbors (FLANN)[5] efficiently handle high-dimensional feature spaces. FLANN's ability to approximate matches using hierarchical trees makes it particularly suited for real-time processing.

While brute force matching is exhaustive and guarantees the best match, its high computational cost limits its applicability in real-time scenarios. FLANN, with its efficient approximation, balances accuracy and speed, making it a preferred choice for practical implementations.

For both brute force and FLANN, Lowe's ratio test is a widely adopted technique to filter out ambiguous matches. By comparing the distance of the best match to the distance of the second-best match, this test ensures that only distinct and reliable matches are retained. This step is essential for ensuring robustness, particularly in noisy environments or when features have multiple potential correspondences.

2.4 Tracking Algorithms

Tracking the movement of features between frames enables accurate speed estimation. Classical tracking algorithms like Lucas-Kanade Optical Flow use local image gradients to estimate motion vectors at a sub-pixel level. However, optical flow methods can be sensitive to lighting changes and fail under large displacements.

Statistical filtering techniques, such as the Kalman Filter[3], are often employed to smooth motion estimates and reject noise. Kalman filters predict the next state of an object based on its motion model, incorporating measurements from new frames to refine predictions. This combination of prediction and correction significantly enhances the reliability of speed estimates in dynamic environments.

2.5 Deep Learning Approaches

In recent years, deep learning has revolutionized object detection and tracking. Models like YOLO (You Only Look Once) have achieved real-time performance with state-of-the-art accuracy in tasks like object detection and motion tracking. YOLOv7[7] offers high-speed accurate predictions.

Incorporating deep learning models like YOLO into speed estimation pipelines could enhance robustness in challenging conditions, such as low-texture environments or dynamic lighting.

The combination of feature detection, matching, tracking, and preprocessing creates a comprehensive pipeline for conveyor belt speed estimation. Feature detection identifies points of interest, matching establishes correspondences between frames, and tracking algorithms estimate motion. Preprocessing ensures that the pipeline remains robust in noisy environments, while Kalman filters refine the results for smooth and reliable speed measurements.

Despite the advances in feature-based methods, challenges like noise sensitivity and dependency on texture remain. Deep learning models offer promising alternatives for future implementations, particularly in scenarios requiring high accuracy and robustness.

3 Methodology

This section discusses the pipeline used in this project.

3.1 Input Data

The input is a 60 FPS stereo video (SVO format) recorded from a ZED camera. Only the left camera's view is considered for the current implementation for simplicity. Figure 1 provides a sample frame.

Figure 1: Sample Frame

3.2 Pre-processing

After extracting the left camera view, polygon masking is applied to isolate the conveyor belt, reducing noise and irrelevant areas from the frame. This ensures that feature detection focuses solely on the region of interest. The masked region is then converted to a grayscale image, simplifying data processing and enabling faster, more efficient feature extraction as shown in Figure 2. As seen, polygon masking helps in avoiding noisy outliers.

Figure 2: Left: Original frame without masking. Right: Frame after polygon masking, isolating the conveyor belt region for focused analysis.

3.3 Feature Detection and Matching

After preprocessing the frame, the next step is feature detection and matching. The SIFT algorithm is employed to extract keypoints from consecutive frames, followed by the FLANN matcher, which uses Lowe's ratio test to filter matches, retaining only high-confidence correspondences.

3.4 Speed Calculation

Speed calculation involves converting image coordinates to real-world measurements using camera calibration parameters, ensuring accurate spatial interpretation of keypoints. The Euclidean distance between matched points across consecutive frames is then computed to determine the displacement. Finally, the frame-to-frame distances, combined with the known frame rate, are used to estimate the speed of the conveyor belt.

3.5 Noise Mitigation

To reduce errors and make speed estimation more accurate, a Kalman Filter is used. This method predicts the movement of the conveyor belt based on previous data and adjusts the prediction using new measurements from each frame. By combining predictions and real data, the Kalman Filter helps to minimize noise and produce smoother, more reliable speed readings.

4 Future Work and Limitations

4.1 Current Limitations

1. **Feature Detector Dependency:** The current implementation relies on SIFT, which is computationally intensive and may struggle in low-texture environments.

2. **Noise Sensitivity:** The current pipeline is susceptible to outliers, especially in environments with dynamic backgrounds.

3. **Stereo Matching Exclusion:** The current implementation does not utilize stereo views, which could improve robustness and accuracy.

4.2 Future Work

1. **Additional Feature Detectors:** Currently the pipeline only uses the SIFT feature detector. In the future, other feature detectors like ORB, SURF, or AKAZE could be incorporated for faster processing and adaptability to different environments.

2. **Deep Learning Models:** As mentioned before, deep learning models offer better robustness. Models such as YOLOv7 can be utilized for robust object detection, while motion-specific frameworks like SORT or DeepSORT can be explored to improve tracking capabilities.

3. **Improved Noise Handling:** Use of more robust statistical methods for outlier rejection (e.g., RANSAC) can be explored.

4. **Stereo Vision:** Current pipeline only utilizes the left view for speed detection. In the future, the pipeline could be extended to leverage both left and right views for redundancy and improved depth estimation.

References

[1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. Similarity Matching in Computer Vision and Multimedia.

[2] David Fernández Llorca, Antonio Hernández Martínez, and Iván García Daza. Vision-based vehicle speed estimation: A survey. *IET Intelligent Transport Systems*, 15(8):987–1005, May 2021.

[3] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.

[4] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004.

[5] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications*, 2009.

[6] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011.

[7] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022.

[8] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, pages 28–31 Vol.2, 2004.