

Article

A Flexible Wheel Alignment Measurement Method via APCS-SwinUnet and Point Cloud Registration

Bo Shi ^{1,2,*} , Hongli Liu ³  and Emanuele Zappa ² 

¹ School of Artificial Intelligence, Changsha University of Science and Technology, Changsha 410114, China

² Department of Mechanical Engineering, Politecnico di Milano, 20156 Milan, Italy; emanuele.zappa@polimi.it

³ College of Electrical and Information Engineering, Hunan University, Changsha 410082, China; hongliliu@hnu.edu.cn

* Correspondence: boshi@hnu.edu.cn

Abstract

To achieve low-cost and flexible wheel angles measurement, we propose a novel strategy that integrates wheel segmentation network with 3D vision. In this framework, a semantic segmentation network is first employed to extract the wheel rim, followed by angle estimation through ICP-based point cloud registration. Since wheel rim extraction is closely tied to angle computation accuracy, we introduce APCS-SwinUnet, a segmentation network built on the SwinUnet architecture and enhanced with ASPP, CBAM, and a hybrid loss function. Compared with traditional image processing methods in wheel alignment, APCS-SwinUnet delivers more accurate and refined segmentation, especially at wheel boundaries. Moreover, it demonstrates strong adaptability across diverse tire types and lighting conditions. Based on the segmented mask, the wheel rim point cloud is extracted, and an iterative closest point algorithm is then employed to register the target point cloud with a reference one. Taking the zero-angle condition as the reference, the rotation and translation matrices are obtained through point cloud registration. These matrices are subsequently converted into toe and camber angles via matrix-to-angle transformation. Experimental results verify that the proposed solution enables accurate angle measurement in a cost-effective, simple, and flexible manner. Furthermore, repeated experiments further validate its robustness and stability.

Keywords: wheel alignment; angle measurement; semantic segmentation; point cloud registration; vision measurement

1. Introduction

Wheel alignment is a routine yet essential maintenance procedure that ensures the wheels are properly angled relative to the vehicle's frame and the road surface [1]. Manufacturers set the wheel alignment within a specified range before the vehicle leaves the factory [2]. However, factors such as suspension wear, heavy loads, poor road conditions, collisions, or chassis damage can cause wheel misalignment, which refers to a deviation from the manufacturer's specifications [3]. Wheel misalignment may involve numerous issues, including uneven tire wear, wheel shimmy, suspension damage, poor handling, increased fuel consumption, and safety risk. Therefore, regular wheel alignment inspections and adjustments are crucial. During maintenance, three primary angles, namely camber, caster and toe angles, are usually measured and adjusted. Toe and camber angles of the vehicle wheels are the two alignment parameters most commonly inspected and adjusted



Academic Editor: Patrice Salzenstein

Received: 21 November 2025

Revised: 23 December 2025

Accepted: 5 January 2026

Published: 12 January 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

during routine maintenance and post-repair operations, and they are also the primary focus of most existing research on vision-based wheel alignment. Therefore, at the current stage, this work primarily focuses on the estimation of toe and camber angles.

1.1. Challenges of 3D Vision-Based Methods

Compared to methods based on inertial measurement unit and 2D vision, the 3D vision-based measurement methods offer advantages such as low cost, high precision, and greater flexibility in application. However, 3D vision-based wheel alignment still faces several challenges, which are summarized below.

- (1) **Complicated calibration and limited flexibility:** Most passive and active 3D measurement systems require a complex and rigorous calibration procedure to reconstruct the wheel shape. Once calibrated, cameras and light sources need to remain fixed, which limits its flexibility and application range. Moreover, the reflective surface of the wheel hub leads to low texture and feature information, which affects stereo matching accuracy and ultimately reduces the accuracy of 3D reconstruction.
- (2) **Additional target board and potential damage:** Commercial scanner-based methods often require mounting a clamped target board with a special reflective film onto the wheel. This process is time-consuming, may cause secondary damage to the wheel, and incurs additional costs due to the consumable film, thereby reducing overall efficiency.
- (3) **Inefficient and noise-sensitive full-cloud registration:** Directly using the entire 3D point cloud acquired by the sensor to estimate wheel angles not only increases computational cost and reduces measurement efficiency, but also introduces interference from background, noise, and points from the vehicle body during point-cloud registration. These irrelevant points can significantly degrade registration accuracy. Therefore, it is more effective to perform 3D registration and angle estimation using only a stable, wheel-related subset of the point cloud.
- (4) **Challenges in precise wheel segmentation:** The reconstructed point cloud includes not only the wheel but also background elements, making accurate region-of-interest (ROI) extraction essential. Existing methods often rely on traditional image processing algorithms (e.g., Canny edge detection and Hough transforms), which are sensitive to background clutter and environmental changes. These methods often require manual parameter tuning and lack the precision needed for accurate wheel boundary and detail extraction.

1.2. Outline of Our Work

To address the challenges outlined above, this paper presents a low-cost and practical wheel alignment method that integrates semantic segmentation with point cloud registration. The proposed approach overcomes the limitations of using 2D or 3D single-modality data alone and fully leverages the rich texture information in 2D images together with the depth and pose information encoded in 3D point clouds, thereby improving both the efficiency and the accuracy of wheel angle measurement. The primary application context of this work is post-production wheel alignment in small and medium-sized garages and workshops, where vehicles are serviced after leaving the factory. The main contributions are summarized as follows.

- (1) As a relatively stable region of the wheel, the wheel rim provides reliable geometric cues, so using only the rim point cloud to estimate wheel angles helps improve both measurement efficiency and angle accuracy. Building on this idea, the proposed method reformulates the wheel alignment task as a pipeline of wheel rim mask extraction, corresponding point cloud registration, and wheel angle calculation.

Compared with existing approaches, it offers greater flexibility, lower cost, and higher efficiency, as it removes the need for complex calibration procedures, target boards, and additional auxiliary equipment or materials.

- (2) Wheel rim extraction is critical for accurate angle estimation. To enhance the precision of wheel rim segmentation, we propose APCS-SwinUnet, a task-driven adaptation of Swin-Unet for wheel rim extraction. Atrous spatial pyramid pooling is embedded into the encoder to capture multi-scale contextual information for wheel rims of different sizes and viewpoints, while a channel-spatial attention mechanism in the decoder selectively enhances rim features and suppress background clutter. This design jointly improves the representation of global wheel contours and fine-grained rim structures, both essential for reliable angle estimation. Compared with traditional image processing pipelines and baseline deep networks, APCS-SwinUnet achieves higher accuracy and robustness in wheel rim segmentation.
- (3) The segmented mask is used to isolate the wheel rim point cloud, effectively filtering out irrelevant background data. The iterative closest point algorithm is then employed to register the initial and target wheel rim point clouds. After registration, the corresponding rotation and translation matrices are obtained, and the rotation matrix is subsequently used to compute the wheel's toe and camber angles.

The rest of this paper is organized as follows. Section 2 reviews related work on wheel alignment. Section 3 presents the overview framework of this paper. Section 4 details the motivation for segmentation and the proposed network structure. Section 5 explains the principle of wheel angles calculated using the iterative closest point method. Section 6 describes the experimental setup and configuration. In Section 7, the segmentation and angle measurement results are presented. Section 8 summarizes the paper and discusses future development.

2. Related Work

In this section, we review related work from two complementary perspectives. First, we summarize existing wheel alignment techniques in terms of sensing and measurement strategies. Most conventional approaches can be broadly grouped into inertial-based methods and vision-based inspection methods, where the latter can be further divided into 2D and 3D vision-based measurement schemes. Second, since the proposed method formulates wheel alignment as a thin structure segmentation problem, we also review recent deep learning-based thin structure segmentation methods.

2.1. Wheel Alignment Sensing and Measurement Methods

2.1.1. Inertial-Based Wheel Alignment Methods

Inertial-based wheel alignment methods typically involve contact measurement. Specific sensors such as inertial measurement units (IMUs), accelerometers, gyroscopes, and angle sensors are mounted on the wheel or vehicle to collect acceleration and angular velocity data [4]. This data is preprocessed to reduce noise, correct sensors bias, and compensate for drift. Afterward, the data is transmitted to a processor (e.g., mobile phone, computer, or embedded device) via Bluetooth, WiFi, or RS232 for further analysis and computation of wheel alignment angles.

Numerous studies have explored the inertial-based methods. Young et al. [4] employed a microcontroller and triaxial accelerometer to measure the camber angle, introducing a calibration technique to correct axis misalignment between the sensor and the measurement system. Chatur et al. [5] proposed a wireless alignment method using accelerometers. D'Mello et al. [6] developed an internet of things (IoT) device consisting of a 6-axis microelectromechanical system (MEMS) sensor (MPU6050), an ESP32 microcontroller, and

battery to measure toe and camber angles. Similarly, Bohari et al. [7] designed a Bluetooth-enabled alignment system using the MPU6050 sensor. Lee et al. [8] presented an online toe angle measurement approach based on lateral tire force, which was validated through simulations in CarSim and Simulink. To reduce energy consumption, Tang et al. [9] introduced a dual wake-up and self-calibration strategy for long-term monitoring. Inertia-based offer methods offer advantages such as high sampling frequency, low latency, low power consumption, and resistance to environmental conditions. However, they face several challenges. Physical contact is often required, risking damage to the wheel hub. Furthermore, sensor drift may cause cumulative errors, and measurement accuracy is affected by electromagnetic interference, temperature, and sensor location.

2.1.2. Vision-Based Wheel Alignment Methods

Compared with inertial-based methods, vision-based wheel alignment methods offer the advantages of convenience, highly automation, and efficiency. Typically, vision-based measurement techniques are divided into 2D and 3D vision measurement methods.

2D Vision-Based Measurement Methods

A 2D vision inspection platform usually consists of target boards, infrared light emitting diodes (IR LEDs), image acquisition modules such as charged coupled device (CCD) or video cameras, and a computer [10]. Each target board, made of reflective material and featuring special patterns (e.g., chessboard or concentric circles), is mounted on the wheel rim using a four-legged clamp to represent the wheel attitude. During the measurement, IR LEDs illuminate the target boards, which are then captured by the CCDs. The technician turns the steering wheel in different directions and moves the vehicle back and forth to determine the axes of rotation and roll for each wheel. Finally, the wheel angles are calculated by analyzing the captured images based on projective geometry. Numerous 2D vision-based methods have been proposed. For instance, Xu et al. [11] proposed a toe and camber angles measurement method using a relevance vector machine and spatial clustering to classify the concentric circles on the target board. Xu et al. [12] proposed a global registration system combining multiple sensors with articulated arms and target boards, although this system is costly and complex. Ge et al. [13] introduced a multi-camera calibration technique using a transparent target board and image correlation. Jiang et al. [14] designed a calibration method for large cabin assembly with an improved homography matrix and combined large targets. Roshan et al. [15] developed a checkboard with augmented markers made of high-emissivity material and aluminum to expand its applicability. Xu et al. [16] proposed a kinpin inclination and caster measurement method using a 1D target with concentric circles.

Two-dimensional vision-based approaches are known for their high accuracy and efficiency, making them widely adopted in vehicle maintenance. However, they also have the following limitations. A complete 2D vision wheel alignment system is expensive, and installing clamps and the target boards is time-consuming and cumbersome, reducing measurement efficiency and increasing costs. Moreover, direct contact between the clamps and the wheels may cause potential.

3D Vision-Based Measurement Methods

Three-dimensional vision-based measurement methods are generally categorized into passive and active 3D vision methods. Passive vision does not project any light source onto the wheel but relies on two or more cameras to recover depth maps or point clouds using triangulation between the camera centers and the wheel. In contrast, active 3D vision involves projecting a structured light source, such as point, line or surface, onto the wheel to reconstruct the object's point cloud. Padegaonkar et al. [17] proposed a binocular

passive vision method to measure wheel angles. Canny edge detection was applied to extract circular contours of the wheel, from which the outermost contour was selected as the region of interest (ROI). Wheel angles were then calculated using the depth map and trigonometry principles. While this method can reconstruct wheel point cloud, it has the following limitations. Passive stereo vision requires stereo rectification and stereo matching. Stereo rectification, based on intrinsic and extrinsic camera parameters, aligns corresponding points onto a common horizontal line. Stereo matching is then performed by exploiting surface textures and feature information. However, wheel hubs are highly reflective and often exhibit homogeneous textures, resulting in limited distinguishable features. This leads to low stereo matching accuracy, which in turn degrades the overall precision of 3D reconstruction. To address these issues, Xu et al. [2] proposed a method using a line laser and 1D target for large-scale onsite measurement. Furferi et al. [18] attached non-structured markers to the tire sidewall, combining image processing and stereo triangulation to associate 3D marker positions with the wheel plane. Senjalia et al. [19] developed a system using laser triangulation and camera calibration, where wheel regions were identified using OpenCV's blob detection. However, this approach requires frequent threshold adjustment due to varying wheel shape, which reduces flexibility and efficiency. Furthermore, lighting conditions, reflections, and surface contamination can significantly affect accuracy. Kim et al. [20] employed two orthogonally mounted laser modules on the wheel knuckle to measure toe and camber angles. Baek et al. [21] developed a surface light-based depth-sensing system that acquires the wheel's point cloud for geometric and posture analysis. Like 2D vision methods, this method requires a target board to be fixed on the wheel. Although no pattern is used, the board is used to represent the attitude of the wheel. To facilitate the ROI extraction, a thin (0.075 mm) adhesive film was applied. Wheel condition was assessed by comparing the current point cloud with a reference plane. While this method enables fast and stable point cloud acquisition, it still requires mounting a target board with a fixed relative position to the wheel. If the relative position changes, an error will be introduced. Moreover, additional matte adhesive film increases costs and reduce efficiency.

2.2. Deep Learning-Based Thin Structure Segmentation Methods

Current research on deep learning-based thin structure segmentation is still relatively limited, and most existing methods are built on convolutional neural networks (CNN) or Transformer-based architectures. Wang et al. [22] proposed a dual-path pavement crack segmentation network that combines a lightweight CNN encoder with a Transformer encoder incorporating attention and an efficient feedforward design to fuse local and global features for crack delineation. Gao et al. [23] proposed a multi-scale global attention network that integrates a densely connected attention U-Net backbone with a global context attention module for hierarchical context aggregation and top-down guidance, achieving consistently competitive retinal vessel segmentation and noticeably improved thin-vessel delineation. Geetha et al. [24] proposed an image-processing-assisted 1D frequency-domain CNN framework that extracts crack candidate regions via adaptive local binarization and then iteratively tracks missing single-pixel-wide propagating cracks. Siriborvornratanakul [25] evaluated Segment Anything Model (SAM) as a zero-shot solution for thin crack segmentation and reported that, although the SAM-Canny pipeline can delineate crack patterns reasonably well on multiple benchmark datasets, its numerical scores remain inferior to task-specific deep learning architectures, e.g., modified U-Net, when assessed under stringent 1-pixel crack annotation criteria. Li et al. [26] developed a dual-path progressive fusion network for retinal vessel segmentation, where a CNN branch is responsible for capturing fine local vessel details and a recurrent convolutional branch models richer contextual information. In addition, they introduced a progressive fusion mechanism

composed of interactive fusion, cross-layer fusion, and scale feature fusion modules to integrate multi-scale features and mitigate the semantic gap between encoder and decoder.

Although these studies have advanced thin structure segmentation for targets such as vessels and pavement cracks, they do not fully address the wheel rim segmentation problem in our wheel alignment pipeline. The wheel rim in this work is not only a thin structure but also exhibits inner and outer boundaries with approximate symmetry, which differs significantly from tree-like or irregular thin patterns, and there is very limited work specifically targeting wheel-like thin structures. Moreover, rim segmentation here is only an intermediate step: our core objective is to estimate wheel angles from the segmented rim point cloud, so background and tire points can affect the robustness of subsequent point cloud matching, making accurate boundary segmentation particularly critical. In addition, practical wheel alignment requires a favorable trade-off between segmentation accuracy and efficiency, whereas many existing thin structure models are too computationally demanding for such scenarios.

In summary, inertial-based methods require physical sensor installation, which may damage the wheel and are sensitive to environmental factors such as electromagnetic interference and temperature. Although vision-based methods improve measurement efficiency, most of them still rely on target boards to estimate the wheel angle. To address these limitations from a technical perspective, this paper proposes a low-cost, non-contact, and practically feasible wheel alignment approach that integrates 3D vision and semantic segmentation techniques. Furthermore, to better handle the wheel rim, which is a thin, wheel-like structure with both inner and outer boundaries and approximate symmetry, we develop an APCS-SwinUNet architecture tailored for accurate rim segmentation.

3. Framework of Proposed Solution

The overall framework of the proposed method is shown in Figure 1. In the first stage, a wheel rim segmentation network based on APCS-SwinUNet is used to extract the wheel rim region from the RGB image. To achieve accurate and fine-grained segmentation, the network incorporates an ASPP module, an attention fusion module, and a hybrid Dice–Focal loss. The resulting segmentation mask is then mapped to the point cloud to obtain the wheel rim point cloud. In the second stage, the initial and target wheel rim point clouds are registered using the ICP algorithm, yielding the rotation and translation matrices that describe the wheel pose. The toe and camber angles are subsequently derived from the rotation matrix and converted to degrees, providing the final wheel angle measurements. Thus, the segmentation network and the point cloud registration are the two core components of the framework, bridging image-level processing and 3D geometric estimation.

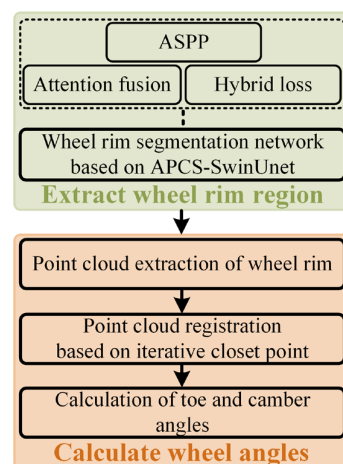


Figure 1. Overall framework of the proposed solution.

4. Wheel Rim Segmentation Based on APCS-SwinUnet

This section presents the network architecture for wheel rim segmentation. We first introduce the motivation and objective of employing a segmentation network for wheel rim extraction. Next, we outline the overall structure of the proposed APCS-SwinUnet, followed by a detailed description of the key components, including atrous spatial pyramid pooling, convolutional block attention module, and hybrid loss function.

4.1. Motivation for Using Segmentation Network

As mentioned in the introduction, most existing wheel alignment methods require mounting a target board on the wheel and involve complex calibration procedures. Although some emerging approaches offer simplified setups, they still rely on target boards or matte adhesive films, which reduce measurement efficiency and increase system cost. To address these limitations, we propose a target-free alignment method by leveraging the point cloud of the wheel rim, eliminating the need for additional auxiliary materials. Traditionally, the target board is used to capture changes in the wheel's attitude and angle due to the non-planar surface of the wheel hub. However, a closer examination reveals that the area between the wheel and the tire contains a circular smooth surface, which naturally spans a range of angles and can effectively reflect the wheel's attitude changes, thereby making it a viable alternative for alignment purposes. The 3D scanner can capture the RGB image and point cloud of the entire scene, whereas our task focuses specifically on extracting the wheel rim's point cloud.

Wheel rim extraction is critical for accurate angle estimation. As shown in Figure 2, the wheel rim is a narrow annular structure that occupies only a small portion of the wheel, making its extraction particularly challenging for the following reasons: (1) The wheel rim has both inner and outer boundaries, and its thin, curved geometry complicates accurate localization. (2) Although there is a noticeable reflectance difference between the wheel rim and the tire, the wheel rim and hub often share similar reflectivity, leading to weak grayscale contrast and ambiguous edges. (3) Since the point cloud of the wheel rim is used to estimate the overall wheel posture, background and tire regions need to be excluded, requiring precise segmentation of the rim contours. Traditional image processing methods can, in principle, perform this task, but they typically involve multiple steps and require manual tuning for different wheel types, resulting in limited flexibility and poor generalization. Deep learning-based segmentation offers a more robust and adaptive alternative.



Figure 2. Schematic illustration of the wheel rim.

Prior studies on thin or narrow structures (such as vessels, roads, or cracks) often leverage multi-scale context and attention mechanisms to improve boundary delineation, and are predominantly implemented within U-Net-like or transformer-based encoder-decoder architectures. However, these methods are generally designed for tree-like or line-like topologies, rather than for closed, narrow annular rims with dual inner-outer boundaries and weak contrast between the rim and the hub, and they are not specifically

optimized for downstream wheel alignment tasks. As a result, existing architectures still struggle to accurately delineate the thin boundary regions of the wheel rim and to provide sufficiently stable contours for subsequent 3D point cloud registration and angle estimation. To address these issues, we design APCS-SwinUnet as a task-specific adaptation of established segmentation modules, tailored to the geometric and photometric characteristics of the wheel rim and its role in wheel alignment measurement.

4.2. Network Architecture of APCS-SwinUnet

The architecture of APCS-SwinUnet is illustrated in Figure 3. It is built upon the Swin Transformer and U-Net frameworks and achieves high segmentation accuracy while maintaining efficiency. First, an atrous spatial pyramid pooling (ASPP) module is added after the third encoder layer to capture multi-scale features relevant to wheel rims. Second, a convolutional block attention module (CBAM) is integrated into the third decoder layer to strengthen the network’s focus on the rim contour and suppress background clutter. Finally, a hybrid loss function combining Dice and Focal losses is employed to better handle class imbalance and stabilize training. In this way, APCS-SwinUnet is not a completely novel architecture, but a carefully engineered combination and adaptation of known modules tailored to the challenges of wheel rim segmentation.

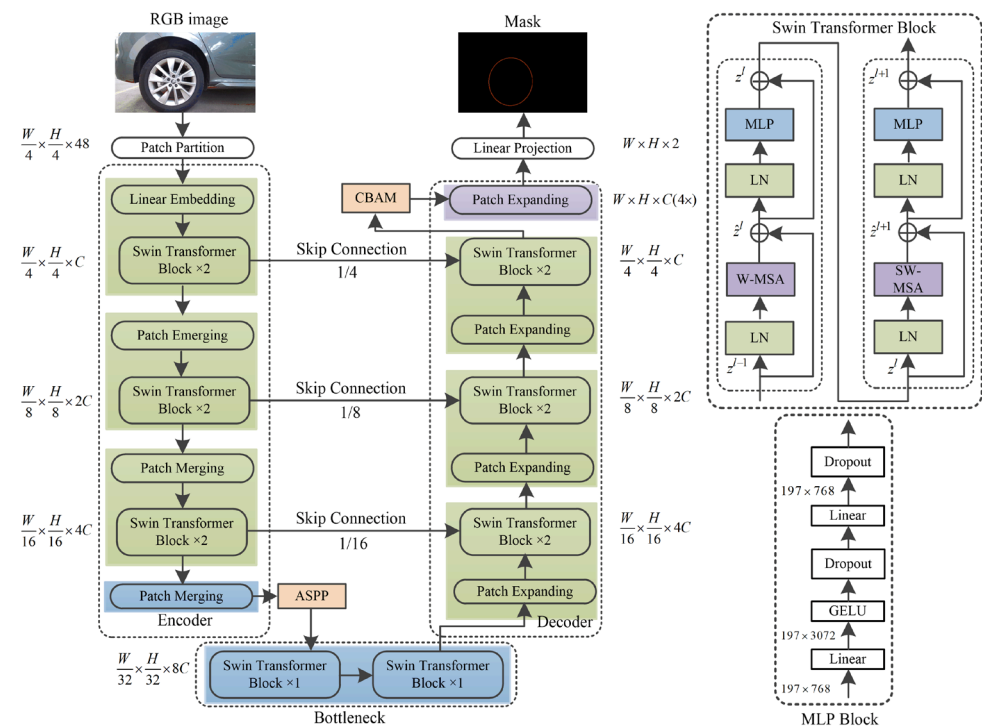


Figure 3. The network architecture of APCS-SwinUnet.

These design choices are particularly effective for wheel rim segmentation for several reasons. The ASPP module provides annulus-aware multi-scale context, enabling the network to capture both the thin rim band and the full circular structure, which helps maintain a closed, consistent rim contour while suppressing spurious responses from the tire and background. CBAM is especially beneficial under weak rim-hub contrast: by refining channel and spatial attention at decoder stages where rim and hub features are mixed, it enhances subtle structural cues along the rim boundary and reduces confusion between these two regions. The Swin-Transformer backbone contributes by modeling long-range dependencies along the rim circumference and across the whole wheel, which promotes global

shape consistency while preserving local boundary details that are crucial for producing stable contours for subsequent 3D point cloud registration and angle estimation.

4.2.1. Atrous Spatial Pyramid Pooling

The segmentation task in this study aims to accurately localize the global contour of the wheel rim while capturing fine-grained boundary details. To enhance segmentation performance, we incorporate the ASPP module, as illustrated in Figure 4. ASPP strengthens the network’s ability to capture multi-scale contextual information by combing atrous convolution with image-level features at varying sampling rates. The ASPP structure [27] consists of multiple convolution layers, atrous convolution layers with different dilation rates, pooling layers, ReLU activations, and an upsampling layer. Its core component is a set of parallel atrous convolutions, which enable the network to extract features across multiple receptive fields without increasing the size or computational cost. Specifically, we employ three parallel atrous convolutions with dilation rates of 4, 6 and 8. Since the proposed segmentation network is based on a Transformer architecture, the input sequence needs to be converted into a 2D convolutional format before entering the ASPP. After processing, the ASPP output is then reshaped back into sequence form for subsequent Transformer-based operations.

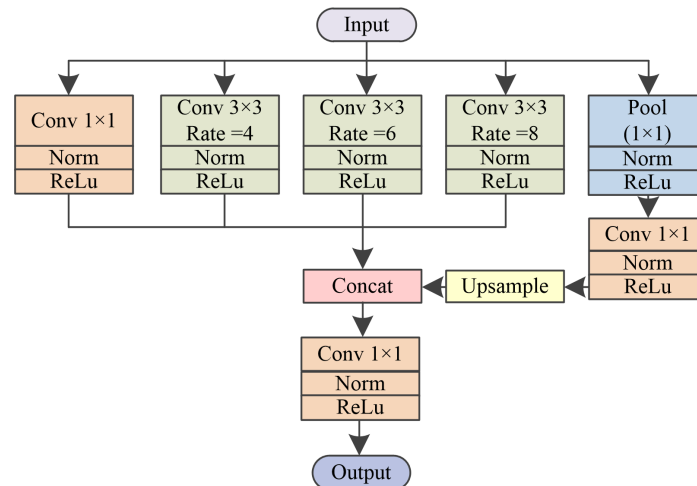


Figure 4. The network architecture of ASPP.

4.2.2. Attention Fusion Module

To enhance the network’s ability to recognize wheel contour, we add a convolutional block attention module after the last Swin-transformer block of the decoder. The structure of the CBAM is shown in Figure 5. The CBAM [28] consists of a channel attention module and a spatial attention module. The channel attention focuses on identifying what features are important, and comprises an average pooling layer, a max pooling layer, a multilayer perceptron (MLP), and a Sigmoid activation. This process is formulated in Equation (1). In contrast, the spatial attention module emphasizes where the important features are located. It includes average and max pooling operations followed by a convolution layer, as described in Equation (2). The final feature map is obtained by sequentially applying both attention mechanisms, as shown in Equation (3).

$$F_c = \sigma \left\{ W_1 \left[W_0 \left(F_{avg_pool} \right) \right] \right\} + W_1 \left[W_0 \left(F_{max_pool} \right) \right] \tag{1}$$

$$F_s = \sigma \left[f^{7 \times 7} \left(F_{avg_pool}, F_{max_pool} \right) \right], F_s \in \mathbb{R}^{1 \times H \times W} \tag{2}$$

$$F' = F_c \otimes F, F'' = F_s \otimes F' \tag{3}$$

where F_{avg_pool} and F_{max_pool} denote the average pooling and maximum pooling, respectively. W_0 and W_1 denote the weights of the MLP. $f^{7 \times 7}$ denotes the convolutional operation with a convolutional kernel size 7×7 . σ denotes the sigmoid function. \otimes represents the element-wise multiplication operation. F' and F'' denote the temporary output and the refined output, respectively. F , F_c and F_s denote the original feature map, output of channel attention module and output of spatial attention module, respectively.

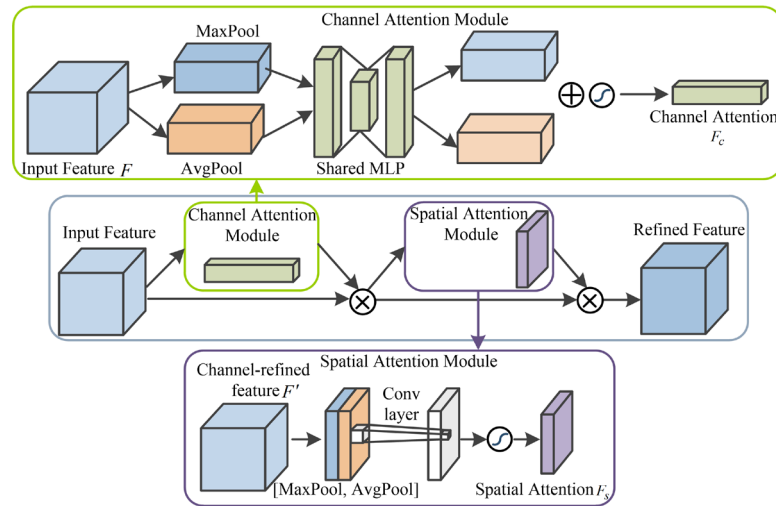


Figure 5. The architecture of the CBAM.

4.2.3. Hybrid Loss Function

To enhance the network’s ability to capture fine details of wheel boundaries and improve training stability, we employed a hybrid loss function that combines Dice loss and Focal loss, as defined in Equations (4)–(7).

$$L_{loss} = (1 - \tau)L_{dice} + \tau L_{focal} \tag{4}$$

$$L_{dice} = \frac{1}{N_D} \sum_{t=1}^{N_D} \left(1 - \frac{2 \times \sum_k p_t^k g_t^k}{\sum_k p_t^k + \sum_k g_t^k} \right) \tag{5}$$

$$L_{focal} = \frac{1}{N_D} \sum_{t=1}^{N_D} \sum_{k=1}^{N_P} -\alpha_t (1 - c_t^k)^{\beta_t} \log(c_t^k) \tag{6}$$

$$c_t^k = \begin{cases} p_t^k, & \text{if } g_t^k = 1 \\ 1 - p_t^k, & \text{if } g_t^k = 0 \end{cases} \tag{7}$$

where L , L_{dice} and L_{focal} denote the total Dice, Dice loss and Focal loss, respectively. τ in Equation (4) is used to balance the contributions of L_{dice} and L_{focal} . p_t^k and g_t^k denote the predicted and ground-truth probabilities for pixel k in category t , respectively. N_D and N_P denote the number of classes and the total number of pixels, respectively. α_t and β_t are introduced to adjust the weighting for each class t , helping to mitigate the class imbalance between foreground and background.

5. Toe and Camber Angles Calculation Based on Iterative Closest Point

This section is about how to combine the segmented mask and the acquired point cloud to calculate the wheel angles. It is organized in three main parts: point cloud extraction of wheel rim, point cloud registration of wheel rim based on iterative closest point and wheel angle calculation.

5.1. Point Cloud Extraction of Wheel Rim

Based on the mask generated by the APCS-SwinUnet, the wheel rim needs to be extracted from the 3D point cloud. As illustrated in Figure 6, the 3D scanner stores point cloud data in a row-wise sequence. Let the point cloud sequence be denoted by $P_i (X_i, Y_i, Z_i)$, where i denotes the index of the point cloud, and X, Y and Z have the same index. The point cloud corresponding to the point (x_w, y_h) on the mask can be calculated using the following equation.

$$\begin{cases} P'_i = P_i, & \text{if } \text{mask}(x_w, y_h) > 0 \\ P'_i = 0, & \text{if } \text{mask}(x_w, y_h) \leq 0 \end{cases}, i = (y_h - 1) \times W + x_w \quad (8)$$

where P and P' denote the original point cloud and the extracted point cloud, respectively. (x_w, y_h) denotes a point on the segmented mask. W and H refer to the width and height of the image, respectively.

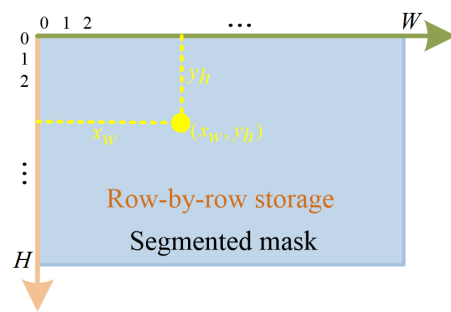


Figure 6. Schematic diagram of point cloud storage.

5.2. Point Cloud Registration of Wheel Rim Based on Iterative Closest Point

This section is about how to combine the segmented mask and the acquired point cloud to calculate the wheel angles. It is organized in three main parts: point cloud extraction of wheel rim, point cloud registration of wheel rim based on iterative closest point and wheel angle calculation. Suppose that the initial and target point cloud sequences are $P\{p_1, p_2, p_3, \dots, p_i\}$, $i \leq N$ and $Q\{q_1, q_2, q_3, \dots, q_j\}$, $j \leq N$, respectively, where N denotes the total number of points. Since P and Q are acquired from different viewpoints by the same 3D scanner, each point j in the set P can find a corresponding point in Q by the rigid transformation, including rotation matrix \mathbf{R} and translation matrix \mathbf{t} , such that

$$\forall_j, q_j = \mathbf{R}p_j + \mathbf{t} \quad (9)$$

Then, we use the iterative closest point (ICP) algorithm [29] to find \mathbf{R} and \mathbf{t} that minimize the alignment error defined by the following objective function.

$$F = \sum_{j=1}^N \|q_j - (\mathbf{R}p_j + \mathbf{t})\|^2 \quad (10)$$

To ensure robustness, a least-squares optimization is performed based on all point correspondences and the singular value decomposition (SVD) algorithm is used to solve for \mathbf{R} and \mathbf{t} . This approach guarantees a stable and optimal solution under the rigid transformation assumption, while ensuring that the resulting \mathbf{R} and \mathbf{t} satisfy orthogonality constraints. Such a formulation not only minimizes the geometric alignment error but also ensures the reliability of rotation matrix \mathbf{R} and translation matrix \mathbf{t} for subsequent angular measurements or geometric analyses.

5.3. Toe and Camber Angles Calculation

After applying the ICP algorithm, the rotation matrix \mathbf{R} is obtained, which describes the rotational relationship between the initial and target point clouds with respect to each axis. The rotation matrix can be expressed as follows:

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \begin{pmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{pmatrix} \quad (11)$$

where $r_{11} \sim r_{33}$ denote the elements of the rotation matrix \mathbf{R} . ψ , θ and ϕ represent the Euler angles around the z , y and x axes, respectively, which are commonly referred to as the yaw, pitch, and roll angles. Since the rotation matrix \mathbf{R} is obtained through the ICP algorithm, the Euler angle can be calculated using Equations (12)–(14).

$$\psi = \tan^{-1} \left(\frac{r_{21}}{r_{11}} \right) \quad (12)$$

$$\theta = -\sin^{-1}(r_{31}) \quad (13)$$

$$\phi = \tan^{-1} \left(\frac{r_{32}}{r_{33}} \right) \quad (14)$$

Then, the Euler angle can be converted from radians to degrees using Equation (15).

$$\psi' = \psi \times \frac{180}{\pi}, \theta' = \theta \times \frac{180}{\pi}, \phi' = \phi \times \frac{180}{\pi} \quad (15)$$

where ψ' , θ' and ϕ' denote the Euler angles in degrees corresponding to ψ , θ and ϕ respectively. In this paper, ϕ' and θ' represent the toe angle and camber angle, respectively.

6. Experimental Setup and Configuration

This section presents the detailed parameters of the measurement system, including the overall system architecture, detailed parameters of the 3-D scanner, clinometer and server.

6.1. Measurement System

As shown in Figure 7, the measurement system consists of an Azure Kinect, a clinometer, a tripod, and a computer. The Azure Kinect was positioned approximately 70 cm from the measured wheel and 29.3 cm above the ground. The intrinsic parameters (focal length, principal point, and lens distortion) and the extrinsic transformation between the RGB and depth cameras are taken from the factory calibration provided by the Azure Kinect SDK. Since the wheel-angle estimation is performed directly in the depth coordinate frame, the relevant extrinsic relationship is the pose of the Azure Kinect with respect to the wheel, defined by the measured distance, height, and the rotation angle given by the clinometer.

Since a commercial wheel alignment system is very expensive, we do not yet have sufficient budget to purchase such equipment. As a result, we adopt a feasibility-oriented experimental setup in which the Azure Kinect sensor is mounted on a tripod with a lockable pan-tilt head, and its pose is adjusted manually via the handle. In this setup, the camber and toe angles are simulated by rotating the Azure Kinect vertically and horizontally, respectively, while keeping the wheel itself stationary. Because the sensor is adjusted manually, it is only approximately, but not perfectly, parallel to the wheel plane. The corresponding camber and toe angles are measured using a clinometer, and these readings are treated as ground-truth values in our experiments. In the implemented system, the initial scan is taken as the geometric reference configuration of the wheel rim, and all subsequent measurements are obtained by registering the new rim point clouds to this reference using ICP. Consequently, the proposed method relies on relative pose estimation

between the reference and current rim configurations, rather than on any absolute pose of the sensor. This design makes the approach inherently robust to non-parallelism between the camera plane and the wheel surface, as long as the relative motion between the sensor and the wheel is accurately captured.



Figure 7. Schematic diagram of the measurement platform.

The main uncertainty sources in this feasibility setup include depth-measurement noise, the limited resolution and alignment accuracy of the clinometer, manual sensor positioning, and segmentation errors in the wheel-rim boundary that propagate to the estimated angles. A complete uncertainty budget and a more detailed repeatability analysis based on higher-precision fixtures and encoders are left for future work.

6.2. Three-Dimensional Scanner Description

The 3D scanner used in this study is the Azure Kinect, as shown in Figure 8. The Kinect is a low-cost 3D scanner that integrates three main imaging components: an RGB camera, a depth camera, and infrared (IR) emitters. It recovers the depth map and 3D point cloud based on the time-of-flight (ToF) principle. Specifically, the device emits IR light pulses onto the object, and the depth is calculated based on the time taken for the reflected light to return to the sensor. The key parameters used in this study are as follows: the RGB camera acquires images at a resolution of 1920×1080 pixels, with a frame rate of 30 fps and a 16:9 aspect ratio. The depth camera operates in Wide Field of View (WFOV) mode. The IR settings include 9 pulses, 125 μs pulse width, 8 idle periods, 1450 μs idle time, and 12.8 ms exposure time. Data acquisition was implemented using the Azure Kinect SDK in Visual Studio 2019. A custom program was developed to capture both RGB images and the corresponding 3D point clouds of the measured wheel.



Figure 8. Schematic diagram of the Azure Kinect.

6.3. Description of Clinometer

The clinometer is used to measure the rotation angles of the Azure Kinect. As shown in Figure 7, it is fixed to the Kinect and rotates together with it either horizontally or vertically. The clinometer used in this study has a measurement range from 0° to 360° degrees, with a resolution of 0.05°.

6.4. Server Configuration

The APCS-SwinUnet network was trained on a server equipped with an NVIDIA GeForce RTX 3090Ti graphics card (24 GB; NVIDIA Corporation, Santa Clara, CA, USA), an Intel Core i9 CPU (4.3 GHz; Intel Corporation, Santa Clara, CA, USA), and 64 GB of RAM (Samsung Electronics Co., Ltd., Suwon, Republic of Korea). The system runs on Ubuntu 20.04. The software environment includes Python 3.8, Pytorch 1.13, and CUDA 11.6.

7. Experimental Results and Analysis

This section aims to verify and evaluate the feasibility of the proposed solution, focusing on the segmentation results and the calculated wheel angles. The experimental details and corresponding results are presented below.

7.1. Wheel Segmentation Experiments

We first provide details of the network training process, including the dataset composition and parameter settings of the proposed APCS-SwinUnet. Subsequently, the segmentation performance is evaluated under normal conditions through both quantitative analysis based on defined evaluation metrics and quantitative assessment of the segmentation results. Next, we evaluate the segmentation performance of the proposed method on our wheel dataset under varying illumination conditions. Then, feature maps from both the encoder and decoder are visualized. Finally, the performance of different segmentation methods is compared on a public wheel dataset.

7.1.1. Datasets for Training and Testing

In this study, 535 different wheel types were collected. These wheel types come from different vehicles, most of which are passenger vehicles, and were captured primarily under sunny and overcast outdoor conditions. Among these 535 wheel types, 423 types were assigned to the training set and the remaining 112 types to the test set. This split is performed at the wheel-type level, so that images (original or augmented) belonging to the same wheel type never appear in both the training and test sets. To enhance the diversity and robustness of the training data, we applied a series of data augmentation techniques to the initial training set, including brightness and contrast adjustment, noise injection, rotation, and other geometric and photometric transformations. As a result, the training set contains 5499 images in total. The test set comprises 112 original, unaltered images and is not augmented, in order to preserve the integrity and validity of the evaluation.

7.1.2. Parameter Settings of APCS-SwinUnet

The network was trained for 30 epochs with a batch size of 6. Input images were resized to 512×512 pixels after being fed into the network. The Adam optimizer was used, with an initial learning rate 0.2 and a drop path rate of 0.001, respectively. The window size and embedding dimension were set to 32 and 96, respectively. The numbers of multi-head attention modules in each stage were 3, 6, 12, and 24. The dilation rates in the ASPP module were set to 4, 6 and 8, respectively.

7.1.3. Evaluation Criteria for Segmentation Network

We first use the Dice similarity coefficient (Dice) to evaluate the overlap and similarity between the predicted mask (P) and the ground truth mask (G), as illustrated in the schematic diagram in Figure 9a. The Dice coefficient can be calculated by Equation (16).

$$Dice = \frac{2|P \cap G|}{|P| \cup |G|} \tag{16}$$

where $|P \cap G|$ and $|P| \cup |G|$ denote the number of intersecting pixels and the total number of pixels of the predicted and ground truth regions, respectively. In addition, we introduce the Hausdorff distance (HD) metric, whose schematic diagram is shown in Figure 9b. While Dice coefficient is more sensitive to the interior of the mask, HD focuses on boundary differences. The HD is defined by the following equation.

$$\begin{cases} D(G, P) = \max(d(G, P), d(P, G)) \\ d(G, P) = \max_{g \in G} \left\{ \min_{p \in P} \|g - p\| \right\} \\ d(P, G) = \max_{p \in P} \left\{ \min_{g \in G} \|p - g\| \right\} \end{cases} \tag{17}$$

where $\|\cdot\|$ represents a distance paradigm operation (e.g., Euclidean distance). The Hausdorff Distance is a bidirectional metric composed of two one-way distances, $d(G, P)$ and $d(P, G)$. Specifically, $D(G, P)$ computes the distance $\|g - p\|$ between each point g_i in the set of G to the point p_i that is closest to this point g_i . Then, the maximum value of these distances is the value of this distance as the value of $d(G, P)$. Similarly, $d(P, G)$ is computed by the same way. Finally, the bidirectional HD is the largest of the two unidirectional distances $d(G, P)$ and $d(P, G)$, which denotes the maximum degree of mismatch between G and P . In this study, we adopt HD95, which represents the 95% of all point-to-set distances between H and G . HD95 can reduce the effect of outliers and enhance the stability of the evaluation.

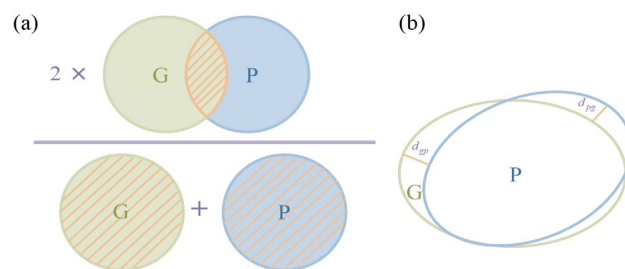


Figure 9. Schematic diagram of metrics. (a) Schematic diagram of Dice. (b) Schematic diagram of HD95.

7.1.4. Comparisons with Different Segmentation Networks

To evaluate the performance of the proposed APCS-SwinUnet network, we conducted comparisons with several representative segmentation networks, including FCN [30], DeepLabV3 [27], OcUnet [31], U-Net [32], Att-Unet [33], U-Net 2+ [34], U-Net 3+ [35], TransUnet [36], and SwinUnet [37] networks. Both quantitative and qualitative segmentation results are provided for comprehensive analysis.

(1) Quantitative Analysis:

The Dice scores achieved by different methods are summarized in Table 1. As shown in Table 1, there are significant differences among the methods. The proposed method achieves a relatively high Dice score and a comparatively low HD95 value. A higher Dice score indicates better overall contour of the wheel, while a lower HD95 indicates improved accuracy in delineating fine boundary details. Empirically, APCS-SwinUnet outperforms

strong baselines, including SwinUnet and TransUnet, in both Dice and HD95. In particular, SwinUnet achieves a Dice of 90.23% and an HD95 of 2.47, whereas APCS-SwinUnet achieves a Dice of 90.66% and an HD95 of 2.11. The 0.43% improvement in Dice reflects better overall rim contour segmentation, while the reduction in HD95 directly indicates more accurate localization of fine boundary regions, which is precisely the aspect targeted in thin-structure segmentation. In practice, these improvements translate into measurable gains in toe and camber angle estimation, confirming that the adapted architecture is effective for the real-world wheel alignment scenario.

Table 1. Comparisons with different networks.

Methods	Dice (%)	HD95
FCN	84.12	8.62
DeepLabV3	86.67	8.18
OcNet	88.11	8.06
U-Net	85.76	11.90
Att-Unet	88.65	9.65
U-Net 2+	89.10	3.00
U-Net 3+	89.49	3.43
TransUnet	89.85	2.56
SwinUnet	90.23	2.47
APCS-SwinUnet	90.66	2.11

These improvements mainly stem from three factors: the strong feature extraction of Swin Transformer blocks, the ASPP module's ability to capture multi-scale rim boundaries, and the CBAM attention mechanism, which focuses on rim regions while suppressing background clutter.

(2) Qualitative Analysis:

To more intuitively compare the strengths and weakness of each method, we conducted a qualitative analysis on typical segmentation results. Specifically, five types of wheels (T1~T5), shown in the first row of Figure 10, were segmented using different network in Table 1. For wheel type T1, all listed methods achieve satisfactory segmented results. However, when the wheel rotates, as in the T2 column, the segmentation becomes more challenging. As shown in the T2 column of Figure 10, U-Net and Att-Unet networks not only miss part of the wheel region but also mistakenly classify background regions as wheel areas. In contrast, the other methods demonstrate better performance on T2 type of wheel. For the T3 type, methods including FCN, DeepLabV3, U-Net and U-Net2+ tend to misclassify the wheel fender as part of the wheel region, likely due to the fender's circular shape resembling the upper portion of the wheel. In type T4, the wheel's lower edge is very close to the ground, making it difficult to distinguish from the background. As a result, methods such as U-Net, Att-UNet, U-Net2+, U-Net3+, and SwinUnet misclassify parts of the ground as the wheel. Type T5 presents a different challenge: the wheel edges are particularly narrow, which leads to partial omission of the wheel region by U-Net, Att-UNet, U-Net2+, U-Net3+, TransUnet, and SwinUnet. In contrast, the proposed method consistently delivers accurate segmentation across all wheel types (T1~T5), producing results that are visually closer to the ground truth (GT), as seen in Figure 10. It is worth noting that although some methods may not show obvious omissions or misclassifications, their segmentation masks may still be slightly larger or smaller than the ground truth, particularly around the boundaries. This discrepancy is reflected in the quantitative results presented in Table 1. In summary, from both quantitative and qualitative perspectives, the proposed method demonstrates superior and more stable segmentation performance

compared to other mainstream networks. Although the Dice score of our method is 0.43% higher than that of SwinUnet, it achieves better boundary precision and segmentation stability, which is beneficial for subsequent point cloud registration and angle computation.

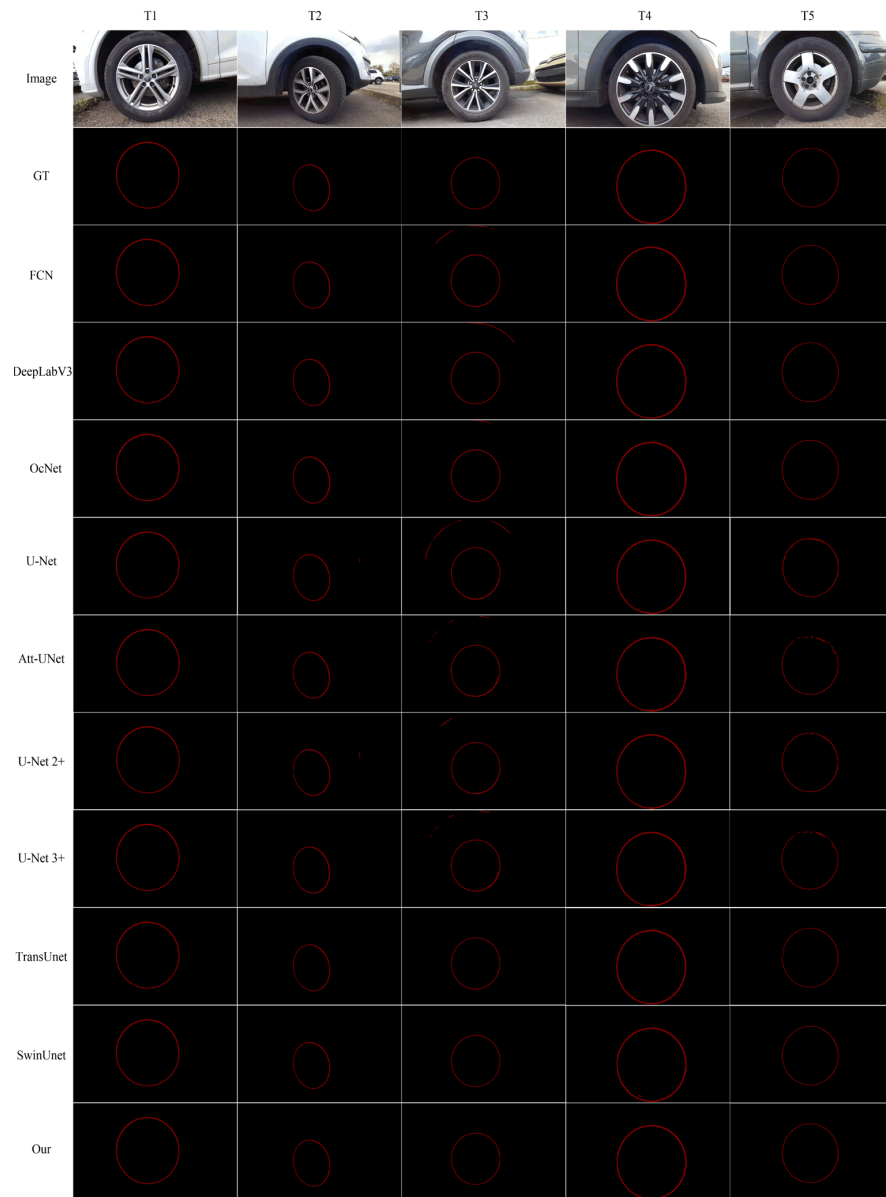


Figure 10. The segmented results produced by different networks.

7.1.5. Comparison Results Under Inconsistent Illuminations

To evaluate the impact of illumination changes, we collected 106 wheels under diverse lighting conditions, including cloudy weather, direct sunlight, and shadow occlusion, with examples shown in Figure 11. The previously trained models were applied to this dataset, and the segmentation results are summarized in Table 2. A comparison between Tables 1 and 2 reveals that most methods suffer from performance degradation to varying degrees under inconsistent illumination, particularly in accurately delineating the edges of the wheel rims, as reflected by the HD95 metrics. In contrast, the proposed method exhibits a smaller performance drop, indicating greater stability and robustness under different lighting conditions.

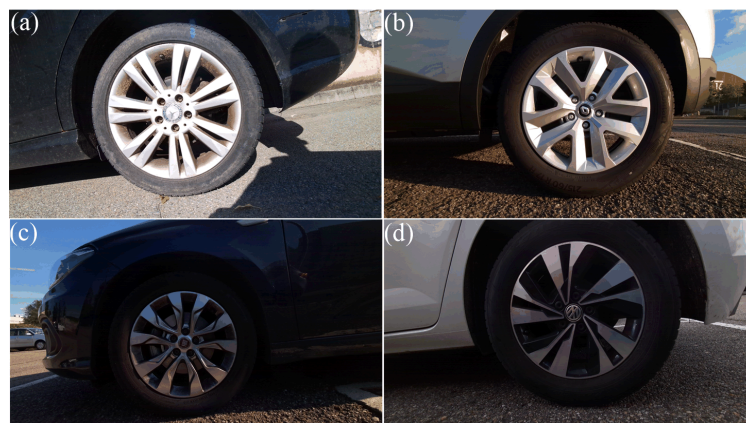


Figure 11. Wheel images under various illumination conditions. (a) Sunlight at noon. (b) Afternoon sunlight. (c) Backlit with shadow. (d) Partial shadow.

Table 2. Comparisons under inconsistent illumination.

Methods	Dice (%)	HD95
FCN	84.35	16.95
DeepLabV3	86.57	10.51
OcNet	87.74	17.48
U-Net	84.10	21.48
Att-UNet	86.97	31.55
U-Net 2+	86.89	25.86
U-Net 3+	88.60	12.46
TransUnet	90.19	2.80
SwinUnet	90.16	2.71
APCS-SwinUnet	90.46	2.51

7.1.6. Feature Visualization of Encoder and Decoder

To provide a clearer and more intuitive understanding of the proposed network’s internal representations, we use the wheel image in Figure 11b as an example and visualize the feature maps of each stage of the encoder and decoder. The feature maps of the four transformer modules in the encoder are shown in Figure 12a–d, while those in the decoder are shown in Figure 12e–h, respectively. For consistency, all feature maps in Figure 12a–h are displayed at the same size. Figure 12a–d illustrate the progressive abstraction from low-level to high-level semantic features, whereas Figure 12e–h demonstrate the gradual restoration of spatial resolution via upsampling. As observed in Figure 12, the APCS-SwinUnet network consistently focuses on regions near the wheel rim in both encoder and decoder stages, while background and vehicle body regions are significantly suppressed. This facilitates accurate segmentation of the wheel rim region, particularly its contour.

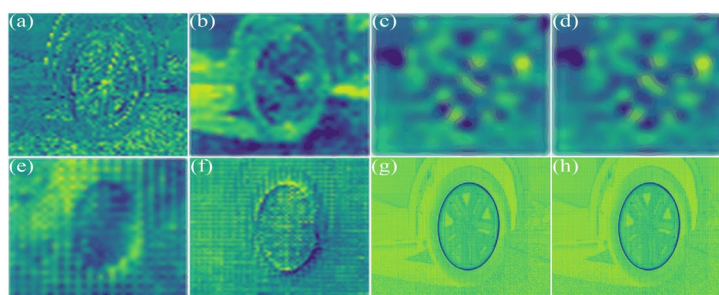


Figure 12. Visualization of feature maps at different stages of the network. (a–d) Encoder stages 1~4 (left side of Figure 3). (e–h) Decoder stage 5~8 (right side of Figure 3).

7.1.7. Comparison Results on the Public Wheel Dataset

To evaluate the performance of the proposed method on public wheel datasets, we selected a publicly available dataset named CAWDEC [38]. Representative sample images are shown in Figure 13. As this dataset does not provide ground-truth masks for wheel hub contours, we manually annotated 101 images to support evaluation. Given the time-consuming labeling process, we annotated only 101 images from the CAWDEC dataset. These annotated images were then used to compare the proposed method with several mainstream segmentation approaches. Thus, this experiment mainly serves as a complementary 2D rim-segmentation benchmark and is inherently constrained by the limited number of manually annotated samples and the practical cost of dense wheel-rim labeling. To better assess the generalization ability of the proposed method, we did not retrain the model on the CAWDEC dataset. Instead, the model trained on our own dataset was directly applied to perform segmentation. The segmentation results of different methods are summarized in Table 3.

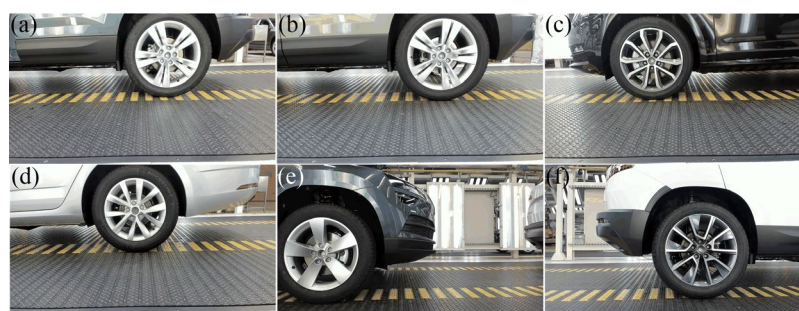


Figure 13. Examples from the public wheel dataset. (a–f) Sample illustrations from various, publicly available wheel datasets.

Table 3. Comparisons of wheel public dataset.

Methods	Dice (%)	HD95
FCN	70.38	28.79
DeepLabV3	72.47	4.37
OcNet	72.24	4.58
U-Net	72.87	16.68
Att-UNet	74.31	6.13
U-Net 2+	74.49	5.53
U-Net 3+	75.50	9.51
TransUNet	75.35	4.00
SwinUNet	75.27	9.92
APCS-SwinUNet	75.88	3.84

As observed, most methods exhibit a notable drop in accuracy, likely due to inconsistencies in acquisition devices, shooting angles, and environmental conditions. Despite these challenges, the proposed method achieves comparatively better segmentation performance. It is worth noting that segmentation serves only as an intermediate step in the proposed framework, with the resulting segmentation masks used to extract the point cloud of the wheel rim. Therefore, all subsequent evaluations are conducted on images captured by the developed platform. As shown in Table 1, the proposed method also achieves strong segmentation performance on the developed platform. Accurate wheel-rim annotation requires fine-grained, pixel-level masks and is therefore extremely time-consuming. Nevertheless, since this research is still in the early stages, we plan to further expand the dataset in both size and diversity in future work.

7.2. Point Cloud Extraction Result of Wheel Rim

After obtaining the wheel rim mask using APCS-SwinUnet, the corresponding point cloud can be extracted from the full point cloud. To illustrate this process, we take the wheel in Figure 14a as an example. Its point cloud, acquired by the Azure Kinect, is shown in Figure 14b. The wheel image is then fed into the APCS-SwinUnet for inference, producing a mask containing only the wheel rim, as shown in Figure 14c. Finally, the point cloud of the wheel rim is extracted using the segmented mask and the method described in Section 5.1, which is shown in Figure 14d.

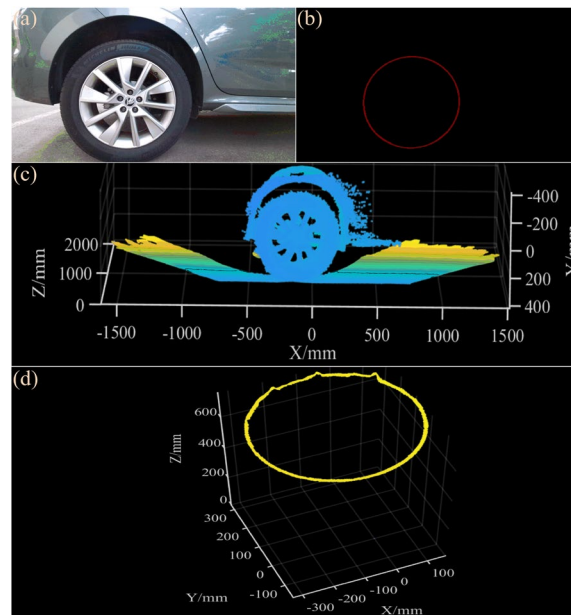


Figure 14. Point cloud extraction results of the wheel rim. (a) RGB image of the wheel. (b) Segmented mask of the wheel rim. (c) Original point cloud. (d) Extracted point cloud corresponding to the wheel rim.

7.3. Wheel Angle Measurement and Evaluation

7.3.1. Point Cloud Registration Result

The initial point cloud and registered point cloud are shown in Figure 15a and 15d, respectively. Figure 15b,c displays the initial point clouds from two additional positions, with their corresponding registration results presented in Figure 15e,f. This study uses the reference and target point clouds shown in Figure 15a as an example to illustrate the process of point cloud registration and wheel angle calculation. The registration between the reference and target point clouds, as shown in Figure 15b, is performed using the ICP algorithm described in Section 5.2. The corresponding rotation matrix is provided in Equation (18).

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{21} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \begin{pmatrix} 0.9988 & 0.0001 & 0.0495 \\ -0.0002 & 1.0000 & 0.0034 \\ -0.0495 & -0.0034 & 0.9988 \end{pmatrix} \quad (18)$$

The rotation matrix \mathbf{R} is an orthogonal matrix used to align the reference point cloud with the target point cloud. The rotation matrix \mathbf{R} contains nine elements, denoted as $r_{11} \sim r_{33}$. The specific definition of $r_{11} \sim r_{33}$ is provided in Equation (11). $\hat{X}(0.9988, -0.0002, -0.0495)^T$, $\hat{Y}(0.0001, 1.0000, -0.0034)^T$ and $\hat{Z}(0.00495, 0.0034, 0.9988)^T$ in matrix \mathbf{R} represent the projections of X' , Y' and Z' axes of the target coordinate system onto the reference coordinate system, respectively. Next, the rotation matrix can be

decomposed into radian angles (ψ, θ, ϕ) representing rotation rotated around Z, Y, X axes. (ψ, θ, ϕ) in radian values are then converted into their corresponding degrees angles (ψ', θ', ϕ'). Both the radian and degree values are summarized in Table 4. Since the Kinect is currently moving along the horizontal direction, this simulated a camber angle, with the resulting angle θ'_y representing the target camber angle.

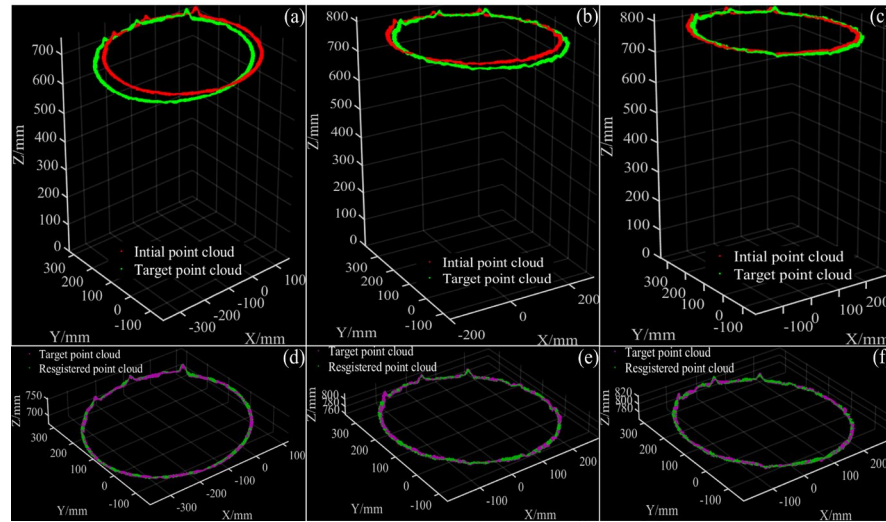


Figure 15. Point cloud registration results between the initial point clouds and target point clouds. (a–c) Initial and target point clouds. (d–f) Corresponding point cloud registration results.

Table 4. Angle measurement results.

ψ (rad)	θ (rad)	ϕ (rad)	ψ' (°)	θ' (°)	ϕ' (°)
−0.00020	0.04952	−0.00340	−0.01146	2.83729	−0.19481

To evaluate the reliability and robustness of point cloud registration process, Gaussian noise with a standard deviation σ ranging from 0.01 to 0.10 was added to both the original reference point cloud and target point cloud shown in Figure 15a. The ICP algorithm was then applied to perform point cloud registration and compute the corresponding rotation matrices, which were subsequently used to calculate the wheel angles. To more intuitively assess the impact of segmentation on angle estimation accuracy, a relative error metric is introduced, as defined in Equation (19).

$$R_e = \frac{|y_i - \hat{y}_i|}{|y_i|} \tag{19}$$

where y_i and \hat{y}_i denote the real angle and the measured angle of i , respectively.

The measured angles, absolute errors and relative errors under different noise levels are summarized in Table 5. As shown in Table 5, the estimated wheel angles remain relatively stable despite increasing noise levels, with both absolute and relative errors being small. These results demonstrate the obtained rotation matrices and the point cloud registration process are stable and robust.

Table 5. Angle measurement results under noisy point clouds.

Noise Level	Real (°)	Meas (°)	Error (°)	R_e (%)
0.01	2.8500	2.8373	−0.0127	0.45
0.02	2.8500	2.8341	−0.0159	0.56

Table 5. *Cont.*

Noise Level	Real (°)	Meas (°)	Error (°)	R_e (%)
0.03	2.8500	2.8333	−0.0167	0.59
0.04	2.8500	2.8334	−0.0166	0.58
0.05	2.8500	2.8339	−0.0161	0.56
0.06	2.8500	2.8348	−0.0152	0.53
0.07	2.8500	2.8340	−0.0160	0.56
0.08	2.8500	2.8333	−0.0167	0.59
0.09	2.8500	2.8339	−0.0161	0.56
0.10	2.8500	2.8336	−0.0164	0.58

7.3.2. Measurement Results of Toe and Camber Angles

In this section, we give the results of angle measurements at different positions and directions. In addition, in order to evaluate the effect of the measurement distance on angle measurement, the measurement platform was placed at different distances from the wheel. The measured results are shown in Tables 6 and 7, respectively. D_s and D_e in Tables 6 and 7 denote the distance of the innermost and outermost sides of the wheel from the platform, respectively. We can see from Tables 6 and 7 that the maximum measurement errors of the toe and camber angles are and -0.105° and 0.158° , respectively. Most of the measured angles are very close to the real angle, which proves that the proposed method has a strong stability with the change in the measurement distance.

Table 6. Measurement results of toe angles.

No	D_s (mm)	D_e (mm)	Real (°)	Meas (°)	Error (°)	R_e (%)
1	687	743	−1.450	−1.457	−0.007	0.48
2	677	754	2.850	2.837	−0.013	0.46
3	691	733	−1.650	−1.755	−0.105	6.36

Table 7. Measurement results of camber angles.

No	D_s (mm)	D_e (mm)	Real (°)	Meas (°)	Error (°)	R_e (%)
1	751	802	−0.750	−0.807	−0.057	7.60
2	778	821	−0.650	−0.685	−0.035	5.38
3	750	808	3.200	3.358	0.158	4.94

7.4. Extended Experiments

7.4.1. Angle Measurement Results of the Raw Point Clouds

To verify the necessity of the proposed segmentation-guided registration scheme, we first attempted direct registration using the raw wheel point clouds, each containing up to 2,073,000 points. However, due to the high point density, point cloud registration was extremely time-consuming and often exhibited poor convergence. To address this issue, we applied downsampling to the raw wheel point clouds at various rates (ranging from 10% to 80%) and performed point cloud registration on the sampled data. The experimental results shown in Table 8 indicate that increasing the sampling rate leads to longer registration times, while the angle estimation accuracy remains limited and does not improve with more point data. This is primarily due to the presence of unstable points from the car body, tire, and background in the raw data, which negatively affect the registration process and ultimately introduce angle estimation errors. In addition to poor accuracy, when the sampling rate reaches 80%, the point cloud registration takes nearly seven hours, rendering the registration process time-consuming and less viable. In contrast, the proposed solution

first segments the wheel rim using a dedicated network before point cloud registration, enabling faster and more accurate angle estimation. This proposed solution not only improves the measurement efficiency but also enhances the angle estimation accuracy. The experimental results in Table 8 validate the above conclusions.

Table 8. Registration time and angle estimation across sampling rates.

Sampling Rate	Time (s)	Real (°)	Meas (°)	R_e (%)
10%	260.105	1.766	−1.084	38.04
20%	1006.079	1.747	−1.103	38.70
30%	2461.575	1.732	−1.118	39.23
40%	5875.771	1.728	−1.122	39.37
50%	8530.951	1.728	−1.122	39.37
60%	10,223.756	1.721	−1.129	39.61
70%	16,907.151	1.720	−1.130	39.65
80%	24,226.577	1.723	−1.127	39.54

7.4.2. Impact of Pixel Shifts on Angle Measurement

To evaluate the impact of wheel rim segmentation on wheel angle estimation, this section uses the wheel mask shown in Figure 16a as an example. The masks are then horizontally shifted by several pixels to the left and right, as shown in Figure 16b,c, respectively. These shifted masks are subsequently used for point cloud registration, and the corresponding wheel angles are calculated. The experimental results are presented in Table 9, where S_L , M_L , S_R , M_R , A_e , and R_e denote the leftward pixel shift, leftward angular measurement, rightward pixel shift, rightward angular measurement, absolute angular error, and relative angular error, respectively. As observed from Table 9, the angle estimation error increases with the magnitude of the pixel shift, regardless of whether the masks are shifted to the left or right. This growing error is primarily attributed to the inclusion of non-wheel-rim point cloud regions in the registration process as the shift increases. These extraneous regions lie outside the actual wheel rim and do not form a consistent planar surface. Their varying distances from the measurement platform introduce inaccuracies in point cloud registration, ultimately leading to deviations from the true wheel angle. These findings highlight the importance of accurate wheel rim segmentation for precise wheel angle estimation.

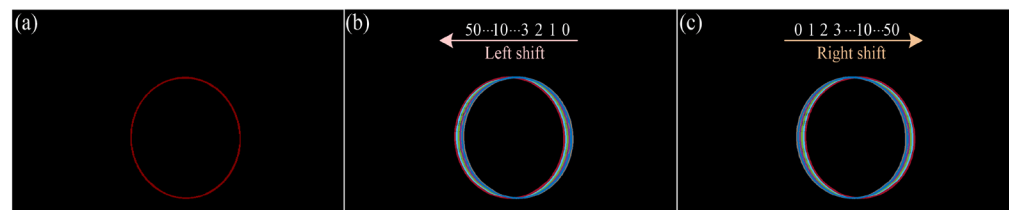


Figure 16. Illustration of pixel-wise mask shifting. (a) Original mask. (b) Left shift in the mask. (c) Right shift in the mask.

Table 9. Angle measurement results under different mask pixel shifts.

S_L	M_L (°)	A_e (°)	R_e (%)	S_R	M_R (°)	A_e (°)	R_e (%)
0	−1.457	0.011	0.48	0	−1.457	0.011	0.48
−1	−1.439	0.011	0.76	1	−1.469	−0.019	1.31
−2	−1.424	0.026	1.79	2	−1.481	−0.031	2.14
−3	−1.416	0.034	2.34	3	−1.498	−0.048	3.31
−4	−1.405	0.045	3.10	4	−1.510	−0.060	4.14

Table 9. *Cont.*

S_L	M_L (°)	A_e (°)	R_e (%)	S_R	M_R (°)	A_e (°)	R_e (%)
−5	−1.394	0.056	3.86	5	−1.528	−0.078	5.38
−6	−1.357	0.093	6.41	6	−1.543	−0.093	6.41
−7	−1.321	0.129	8.90	7	−1.560	−0.110	7.59
−8	−1.291	0.159	10.97	8	−1.594	−0.144	9.93
−9	−1.257	0.193	13.31	9	−1.612	−0.162	11.17
−10	−1.202	0.248	17.10	10	−1.666	−0.216	14.90
−15	−0.840	0.610	42.07	15	−2.019	−0.569	39.24
−20	−0.480	0.970	66.90	20	−2.431	−0.981	67.66
−30	0.207	1.657	114.28	30	−3.101	−1.651	113.86
−40	0.963	2.413	166.41	40	−3.833	−2.383	164.34
−50	1.850	3.300	227.59	50	−4.892	−3.442	237.38

7.4.3. Impact of Segmentation Methods on Angle Measurement

To evaluate the impact of different segmentation methods on wheel angle measurement, a wheel with a ground-truth angle of 3.200° is selected as an example for analysis. Segmentation masks generated by various methods are used to estimate the wheel angle, and the results are summarized in Table 10. As shown in Table 10, the accuracy of angle estimation varies significantly across different segmentation methods, primarily due to significant differences in the Dice coefficients and HD95 values. Compared with the other methods, the proposed approach achieves more accurate segmentation of the wheel rim, particularly around the boundaries, which results in higher precision in the final angle estimation. The experimental results presented in this section demonstrate a strong correlation between the accuracy of wheel rim segmentation and the accuracy of angle estimation. Moreover, compared with other methods, the proposed approach yields smaller angle estimation errors, further validate its effectiveness.

Table 10. Angle measurement results of different segmentation methods.

Methods	Dice (%)	HD95	Meas (°)	Error (°)	R_e (%)
FCN	83.29	2.24	3.425	0.225	7.03
DeepLabV3	85.12	3.00	3.530	0.330	10.31
OcNet	86.36	2.24	3.515	0.315	9.84
U-Net	87.94	4.47	3.769	0.569	17.78
Att-Unet	89.67	3.61	3.646	0.446	13.94
U-Net 2+	89.13	4.00	2.741	−0.459	14.34
U-Net 3+	89.27	2.00	2.980	−0.220	6.88
TransUnet	89.72	2.83	2.874	−0.326	10.19
SwinUnet	90.09	2.24	3.404	0.204	6.38
APCS-SwinUnet	90.55	2.00	3.358	0.158	4.94

7.4.4. Runtime and Computational Efficiency

To quantify the computational efficiency of the proposed solution, we evaluated the end-to-end runtime of the complete pipeline, which consists of three main stages: wheel rim edge segmentation, point cloud registration, and angle estimation. On our test platform, the segmentation network requires approximately 0.463 s per image to generate the wheel rim mask. The subsequent point cloud registration and angle estimation take about 4.476 s per sample, resulting in a total processing time of roughly 4.939 s per wheel. For comparison, when point cloud matching is performed directly on the raw point cloud without prior segmentation, the computational burden increases drastically. At a sampling rate of 80%, the point cloud registration alone requires nearly 7 h to complete, and the resulting accuracy is significantly lower. These results demonstrate that the proposed method

offers a favorable trade-off between computational efficiency and practical applicability in wheel-alignment scenarios.

7.4.5. Repeatability Experiments at Different Distances and Illuminations

To evaluate the stability and robustness of the proposed method under varying distances and illuminations, a series of repeated experiments were conducted. Four alignment cases were considered in this section: toe-in, toe-out, negative camber, and positive camber. The experiments were carried out on a sunny and breezy day, with the ambient temperature around 31 degrees Celsius. To account for illumination variations, wheel measurements were taken at 11:39 a.m., 12:45 p.m., 18:01 p.m., and 19:08 p.m. The corresponding wheel images captured at these different times are shown in Figure 17.

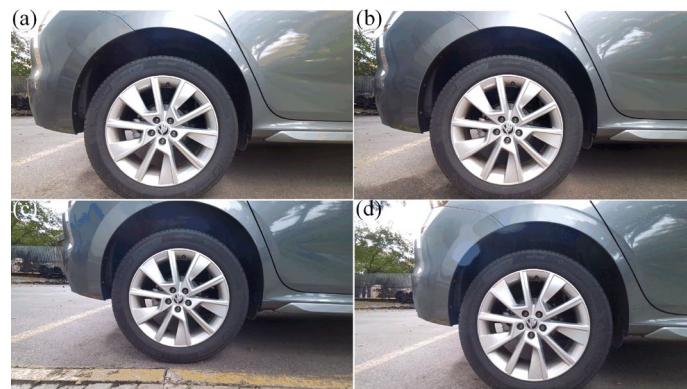


Figure 17. Wheel images under different illumination conditions and distances. (a–d) Wheel images captured at 12:45 p.m., 11:39 a.m., 18:01 p.m. and 19:08 p.m., respectively.

Taking the toe-in case as an example, five RGB images and corresponding point clouds were acquired at the initial wheel position using the Azure Kinect, with a two-minute interval between each capture. After repositioning the sensor, another five sets of RGB images and point clouds were collected at the target position using the same procedure. Including the time required for data storage, each angular repeatability experiments took approximately twenty minutes. Given the extended acquisition period, several random factors were considered, including sensor noise, illumination variation, wind disturbances, and the inherent measurement uncertainty of the Azure Kinect. The repeatability results for toe-in, toe-out, negative camber, and positive camber angles are summarized in Table 11. For each alignment case, five data groups were collected at both the initial and target positions. These were paired to compute five corresponding angular values per case. As shown in Table 11, the angular measurement errors under varying distances and lighting conditions remain within a narrow range. To quantitatively assess the accuracy, mean absolute error (MAE) and root mean square error (RMSE) were calculated according to Equation (20).

$$\begin{cases} MAE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i) \\ RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \end{cases} \quad (20)$$

where *MAE* and *RMSE* denote the mean absolute error and root mean square error. y_i and \hat{y}_i denote the real value and the measured value of i , while N denotes the total number of samples. The MAE and RMSE for the toe angle are -0.045° and 0.078° , respectively, while those for the camber angle are 0.010° and 0.066° , respectively. These repeated experiments show that the proposed method exhibits good stability and robustness against environmental variations. It is worth noting that direct sunlight should be avoided during

measurement. A distance ranging from 670 mm to 850 mm between the wheel and the measurement platform is considered acceptable. Furthermore, since wheel alignment tasks are typically performed indoors in vehicle maintenance facilities, the measurement environment is generally well-controlled and stable.

Table 11. Repeated experiments under different distances and illuminations.

No	Toe-In			Toe-Out			Negative-Camber			Positive-Camber		
	12:45 p.m.–12.59 p.m.			11:39 a.m.–11:57 a.m.			18:01 p.m.–18:19 p.m.			19:08 p.m.–19:25 p.m.		
	$D_s = 698$ mm	$D_e = 744$ mm	Error (°)	$D_s = 704$ mm	$D_e = 742$ mm	Error (°)	$D_s = 769$ mm	$D_e = 831$ mm	Error (°)	$D_s = 698$ mm	$D_e = 745$ mm	Error (°)
Real (°)	Meas (°)		Real (°)	Meas (°)		Real (°)	Meas (°)		Real (°)	Meas (°)		Real (°)
1	-0.900	-0.902	-0.002	1.400	1.307	-0.093	-0.900	-1.014	-0.114	1.250	1.323	0.073
2	-0.900	-0.946	-0.046	1.400	1.292	-0.108	-0.900	-0.932	-0.032	1.250	1.260	0.010
3	-0.900	-0.933	-0.033	1.400	1.227	-0.173	-0.900	-0.954	-0.054	1.250	1.224	-0.026
4	-0.900	-0.877	0.023	1.400	1.283	-0.117	-0.900	-1.000	-0.100	1.250	1.228	-0.022
5	-0.900	-0.912	-0.012	1.400	1.281	-0.119	-0.900	-0.997	-0.097	1.250	1.315	0.065
6	-0.900	-0.873	0.027	1.400	1.348	-0.052	-0.900	-0.935	-0.035	1.250	1.401	0.151
7	-0.900	-0.904	-0.004	1.400	1.311	-0.089	-0.900	-0.887	0.013	1.250	1.325	0.075
8	-0.900	-0.873	0.027	1.400	1.259	-0.141	-0.900	-0.888	0.012	1.250	1.289	0.039
9	-0.900	-0.838	0.062	1.400	1.314	-0.086	-0.900	-0.924	-0.024	1.250	1.287	0.037
10	-0.900	-0.863	0.037	1.400	1.318	-0.082	-0.900	-0.938	-0.038	1.250	1.389	0.139
11	-0.900	-0.906	-0.006	1.400	1.380	-0.020	-0.900	-0.997	-0.097	1.250	1.346	0.096
12	-0.900	-0.959	-0.059	1.400	1.355	-0.045	-0.900	-0.925	-0.025	1.250	1.274	0.024
13	-0.900	-0.927	-0.027	1.400	1.313	-0.087	-0.900	-0.926	-0.026	1.250	1.251	0.001
14	-0.900	-0.870	0.030	1.400	1.358	-0.042	-0.900	-0.958	-0.058	1.250	1.235	-0.015
15	-0.900	-0.906	-0.006	1.400	1.368	-0.032	-0.900	-0.974	-0.074	1.250	1.342	0.092
16	-0.900	-0.873	0.027	1.400	1.258	-0.142	-0.900	-0.958	-0.058	1.250	1.392	0.142
17	-0.900	-0.924	-0.024	1.400	1.256	-0.144	-0.900	-0.891	0.009	1.250	1.316	0.066
18	-0.900	-0.909	-0.009	1.400	1.214	-0.186	-0.900	-0.907	-0.007	1.250	1.304	0.054
19	-0.900	-0.850	0.050	1.400	1.266	-0.134	-0.900	-0.944	-0.044	1.250	1.283	0.033
20	-0.900	-0.899	0.001	1.400	1.279	-0.121	-0.900	-0.948	-0.048	1.250	1.381	0.131
21	-0.900	-0.889	0.011	1.400	1.342	-0.058	-0.900	-0.922	-0.022	1.250	1.343	0.093
22	-0.900	-0.918	-0.018	1.400	1.326	-0.074	-0.900	-0.851	0.049	1.250	1.275	0.025
23	-0.900	-0.902	-0.002	1.400	1.272	-0.128	-0.900	-0.868	0.032	1.250	1.248	-0.002
24	-0.900	-0.842	0.058	1.400	1.331	-0.069	-0.900	-0.907	-0.007	1.250	1.236	-0.014
25	-0.900	-0.877	0.023	1.400	1.342	-0.058	-0.900	-0.909	-0.009	1.250	1.343	0.093
MAE	-0.045						0.010					
RMSE	0.078						0.066					

8. Conclusions and Future Development

To enable low-cost and flexible wheel alignment, this paper proposes a novel strategy that integrates semantic segmentation and 3D vision. We develop an APCS-SwinUnet segmentation network that incorporates ASPP, CBAM, and a hybrid loss function to achieve accurate wheel rim extraction. Using the segmented rim as the region of interest, we apply the ICP algorithm to register the reference and target point clouds, and then convert the resulting rotation matrix into toe and camber angles. Experimental results show that high-quality segmentation is crucial for reliable point-cloud registration and precise angle estimation. Overall, the proposed solution provides accurate, stable, and robust wheel-angle measurements while remaining cost-effective and flexible compared with conventional wheel alignment methods.

Quantitatively, APCS-SwinUnet achieves a Dice score of 90.66% and an HD95 of 2.11 for rim segmentation, representing a 0.43% increase in Dice and a 0.36 reduction in HD95 compared with the baseline model. Repeated experiments further show that the MAE and RMSE are -0.045° and 0.078° for the toe angle, and 0.010° and 0.066° for the camber angle, respectively, confirming that the proposed system can deliver accurate and reliable wheel-angle measurements. In the revised manuscript, we also relate these errors to the accuracy requirements specified in standards for four-wheel alignment instruments. Specifically, the allowable indication errors for total toe and single-wheel toe are within $\pm 4'$, and that for camber is within $\pm 2'$, corresponding to approximately $0.03\text{--}0.07^\circ$. Although some of our current measurements yield absolute toe and camber errors that are slightly larger, the errors remain well below 0.1° and are close to the target accuracy range.

The primary purpose of this study is to validate the feasibility of combining semantic segmentation with point-cloud registration for wheel-angle measurement. Accordingly, the current experiments focus on a single wheel in a controlled environment, and the proposed system should be regarded as a feasibility prototype rather than an industrial-grade product. For applications requiring higher angle-measurement accuracy, the proposed algorithmic framework can, in principle, be integrated with industrial-grade 3D sensors. We also explicitly acknowledge that the RGB images and associated 3D point-cloud data in our self-collected dataset have limited diversity in wheel types and acquisition conditions. This limitation constrains the current assessment of generalization and motivates future extensions to larger and more diverse datasets. In particular, we have outlined a future plan to enlarge the 3D dataset with additional wheel designs, environmental conditions, and viewpoints so as to more comprehensively assess and improve the generalization ability of the proposed model. Due to current resource constraints, a direct quantitative comparison with a commercial wheel-alignment system is not yet feasible. However, we recognize that such a benchmark would be technically sound and highly valuable, and we plan to allocate future resources to acquire a commercial system for more rigorous, industry-relevant validation. Future work will first extend the system to simultaneously measure the angles of all four wheels by integrating four such subsystems. We also plan to further narrow the gap to industrial standards by (i) adopting industrial-grade 3D scanners with higher stability and smaller systematic errors, (ii) performing rigorous comparisons against certified commercial alignment systems, and (iii) conducting a quantitative uncertainty analysis in accordance with metrological standards (e.g., ISO-GUM [39]), including a complete measurement model, an uncertainty budget, and traceability to SI units. Additional research will address robustness under more challenging conditions, such as dirty or partially occluded wheels and diverse rim geometries, and will investigate dedicated mechanical fixtures to improve stability and repeatability. Finally, we envision leveraging advanced machine-learning techniques not only for segmentation but also for data-driven uncertainty estimation, thereby strengthening the connection between this work and broader topics in three-dimensional metrology and machine learning for metrology.

Author Contributions: Conceptualization, B.S. and E.Z.; methodology, B.S. and E.Z.; software, B.S.; validation, B.S., E.Z. and H.L.; formal analysis, B.S., E.Z. and H.L.; investigation, B.S. and E.Z.; resources, E.Z.; data curation, B.S.; writing—original draft preparation, B.S.; writing—review and editing, B.S., E.Z. and H.L.; visualization, B.S.; supervision, E.Z.; project administration, E.Z.; funding acquisition, E.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Nature Science Foundation of China under Grant 62173133, in part by the Postgraduate Study Abroad Program by China Scholarship Council under Grant 202306130094, and in part by the Postgraduate Scientific Research Innovation Project of Hunan Province CX20220398.

Data Availability Statement: The datasets used for the examples described outside Section 7.1.7 are not readily available, because the data are part of an ongoing study. The publicly available dataset [38] was used for the example described in Section 7.1.7.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yunta, J.; Garcia-Pozuelo, D.; Diaz, V.; Olatunbosun, O. Influence of camber angle on tire tread behavior by an on-board strain-based system for intelligent tires. *Measurement* **2019**, *145*, 631–639. [CrossRef]
2. Xu, G.; Wang, Y.; He, W.; Shen, H.; Chen, F.; Li, X.T.; Zhao, X.X. Large-scale all-wheel alignment globally registered by laser line family and verified by global benchmark. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 5013510. [CrossRef]
3. Young, J.S.; Hsu, H.Y.; Chuang, C.Y. Camber angle inspection for vehicle wheel alignments. *Sensors* **2017**, *17*, 285. [CrossRef]

4. Feng, L.H.; Chen, W.; Cheng, M.; Zhang, W.G. The gravity-based approach for online recalibration of wheel force sensors. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 1686–1697. [[CrossRef](#)]
5. Chatur, S. Computer based wireless automobile wheel alignment system using accelerometer. *Inter. J. Eng. Sci.* **2015**, *4*, 62–69.
6. D’Mello, G.; Gomes, R.; Mascarenhas, R.; Ballal, S.; Kamath, V.S.; Lobo, V.J. Wheel alignment detection with IoT embedded system. *Mater. Today Proc.* **2022**, *52*, 1924–1929. [[CrossRef](#)]
7. Bohari, A.A.; Hafiz, M.F.H.M.; Yi, S.S.; Jamal, N.; Talib, M.N.M.; Safuan, S.N.M. Development of automobile wheel smart alignment monitoring system. *PaperASIA* **2024**, *40*, 28–35. [[CrossRef](#)]
8. Lee, H.; Choi, S.B. Online detection of toe angle misalignment based on lateral tire force and tire aligning moment. *Int. J. Automot. Technol.* **2023**, *24*, 623–632. [[CrossRef](#)]
9. Tang, X.L.; Shi, Y.; Chen, B.; Longden, M.; Farooq, R.; Lees, H.; Jia, Y. A miniature and intelligent low-power in situ wireless monitoring system for automotive wheel alignment. *Measurement* **2023**, *211*, 112578. [[CrossRef](#)]
10. Song, L.M.; Wang, R.H.; Chen, E.Z.; Yang, Y.G.; Zhu, X.J.; Liu, M.Y. Research on global calibration method of large-scene multi-vision sensors in wheel alignment. *Meas. Sci. Technol.* **2022**, *32*, 105023. [[CrossRef](#)]
11. Xu, G.; Wei, H.; Fang, C.; Hui, S.; Li, X.T. Automatic and accurate vision-based measurement of camber and toe-in alignment of vehicle wheel. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 5024613. [[CrossRef](#)]
12. Xu, G.; Shen, H.; Li, X.T.; Chen, F.; He, W. Large-range reconstruction with non-pre-calibrated camera via MBM of laser referenced by radial-collinear-features. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 5013510. [[CrossRef](#)]
13. Ge, P.X.; Wang, H.Q.; Wang, Y.H.; Wang, B. Calibration of ring multicamera system with transparent target for panoramic measurement. *IEEE Sens. J.* **2022**, *22*, 23154–23164. [[CrossRef](#)]
14. Jiang, T.; Cui, H.; Cheng, X.S. A calibration strategy for vision-guided robot assembly system of large cabin. *Measurement* **2020**, *163*, 107991. [[CrossRef](#)]
15. Roshan, M.C.; Isaksson, M.; Pranata, A. A geometric calibration method for thermal cameras using a ChArUco board. *Infrared Phys. Technol.* **2024**, *138*, 105219. [[CrossRef](#)]
16. Xu, G.; He, W.; Chen, F.; Shen, H.; Li, X.T. One-dimension orientation method of caster and kingpin inclination of vehicle wheel alignment. *Measurement* **2022**, *198*, 111371. [[CrossRef](#)]
17. Padegaonkar, A.; Brahme, M.; Bangale, M.; Raj, A.N.J. Implementation of machine vision system for finding defects in wheel alignment. *Int. J. Comput. Inf. Technol.* **2014**, *1*, 339–344.
18. Furferi, R.; Governì, L.; Volpe, Y.; Carfagni, M. Design and assessment of a machine vision system for automatic vehicle wheel alignment. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 242. [[CrossRef](#)]
19. Senjalia, J.; Pandya, P.; Kapadia, H. Measurement of wheel alignment using camera calibration and laser triangulation. In Proceedings of the 2013 Nirma University International Conference on Engineering (NUICONE), Ahmedabad, India, 28–30 November 2013; pp. 1–5.
20. Kim, S.H.; Lee, K.I. Wheel alignment of a suspension module unit using a laser module. *Sensors* **2020**, *20*, 1648. [[CrossRef](#)] [[PubMed](#)]
21. Baek, D.; Cho, S.; Bang, H. Wheel alignment inspection by 3D point cloud monitoring. *Mech. Sci. Technol.* **2014**, *28*, 1465–1471. [[CrossRef](#)]
22. Wang, J.; Zeng, Z.; Sharma, P.K.; Alfarraj, O.; Tolba, A.; Zhang, J.; Wang, L. Dual-path network combining CNN and transformer for pavement crack segmentation. *Autom. Constr.* **2024**, *158*, 105217. [[CrossRef](#)]
23. Gao, G.; Li, J.Y.; Yang, L.; Liu, Y.H. A multi-scale global attention network for blood vessel segmentation from fundus images. *Measurement* **2023**, *222*, 113553. [[CrossRef](#)]
24. Geetha, G.K.; Yang, H.J.; Sim, S.H. Fast detection of missing thin propagating cracks during deep-learning-based concrete crack/non-crack classification. *Sensors* **2023**, *23*, 1419.
25. Siriborvornratanakul, T. Image segmentation for thin structures using a zero-shot learner. *Int. J. Inf. Technol.* **2025**, *17*, 721–726. [[CrossRef](#)]
26. Li, J.Y.; Gao, G.; Yang, L.; Bian, G.B.; Liu, Y.H. DPF-Net: A dual-path progressive fusion network for retinal vessel segmentation. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 2517817. [[CrossRef](#)]
27. Chen, L.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587. [[CrossRef](#)]
28. Woo, S.; Park, J.; Lee, J.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
29. Xu, M.B.; Han, Y.M.; Zhong, X.T.; Sang, F.Y.; Zhang, Y.A. A precise registration method for large-scale urban point clouds based on phased and spatial geometric features. *Meas. Sci. Technol.* **2025**, *36*, 015202. [[CrossRef](#)]
30. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.

31. Yuan, Y.H.; Huang, L.; Guo, J.Y.; Zhang, C.; Chen, X.L.; Wang, J.D. OCNet: Object context for semantic segmentation. *Int. J. Comput. Vis.* **2021**, *129*, 2375–2398. [[CrossRef](#)]
32. Zunair, H.; Hamza, A.B. Sharp U-Net: Depthwise convolutional network for biomedical image segmentation. *Comput. Biol. Med.* **2021**, *136*, 104699. [[CrossRef](#)] [[PubMed](#)]
33. Oktay, O.; Schlemper, J.; Folgoc, L.L.; Lee, M.; Heinrich, M.; Misawa, K.; Mori, K.; McDonagh, S.; Hammerla, N.Y.; Kainz, B.; et al. Attention U-Net: Learning where to look for the pancreas. *arXiv* **2018**, arXiv:1804.0399. [[CrossRef](#)]
34. Zhou, Z.W.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. UNet++: A nested U-Net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–11.
35. Huang, H.M.; Lin, L.F.; Tong, R.F.; Hu, H.J.; Zhang, Q.W.; Iwamoto, Y.; Han, X.; Chen, Y.-W.; Wu, J. Unet 3+: A full-scale connected unet for medical image segmentation. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 1055–1059.
36. Chen, J.N.; Lu, Y.Y.; Yu, Q.H.; Luo, X.D.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y. TransUNet: Transformers make strong encoders for medical image segmentation. *arXiv* **2021**, arXiv:2102.04306. [[CrossRef](#)]
37. Cao, H.; Wang, Y.Y.; Chen, J.; Jiang, D.S.; Zhang, X.P.; Tian, Q.; Wang, M.N. Swin-Unet: Unet-like pure Transformer for medical image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 205–218.
38. Available online: <https://www.kaggle.com/datasets/adamnovozmsk/cawdec?resource=download> (accessed on 31 May 2019).
39. ISO/IEC. *ISO/IEC Guide 98-6:2021(en), Uncertainty of Measurement—Part 6: Developing and Using Measurement Models*; International Organization for Standardization: Geneva, Switzerland, 2021.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.