2025 IEEE Intelligent Vehicles Symposium (IV)
June 22-25, 2025. Cluj-Napoca, Romania

Assisted Trailer Parking using a Reverse Camera System and Inverse Kinematics

Daniel Kreimer
Inst. of Visual Computing
Graz Univ. of Technology
Graz, Austria
kreimer@student.tugraz.at

Philipp Fleck
Inst. of Visual Computing
Graz Univ. of Technology
Graz, Austria
philipp.fleck@tugraz.at

Thomas Kernbauer
Inst. of Visual Computing
Graz Univ. of Technology
Graz, Austria
kernbauer@tugraz.at

Clemens Arth
Inst. of Visual Computing
Graz Univ. of Technology
Graz, Austria
clemens.arth@tugraz.at

2025 IEEE Intelligent Vehicles Symposium (IV) | 979-8-3315-3803-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/IV64158.2025.11097420

***Abstract*—We introduce a rear-view camera system designed for trailers, comprising multiple cameras, to address the challenges of limited visibility and the awkward maneuverability of a trailer when performing a reverse driving task (*e.g.,* parking). The cameras are placed on the top edge of the trailer looking downwards, while the different camera feeds are transformed and stitched to create a top-down view. The actual steering angle of the vehicle and the hitch angle are measured using IoT hardware. The path of the trailer is determined using inverse kinematics of the vehicle and trailer, and lines are overlaid on the top-down view to provide drivers with dynamic, real-time guiding lines on their smartphone. We describe a practical implementation of our system on one instance of a vehicle and a trailer, and extrapolate the concept to different types of trailers in simulations.**

***Index Terms*—Augmented Reality, Trailer, Vehicle, Inverse Kinematics, Camera System.**

I. Introduction

Trailers have become an integral part of global logistics and are essential for different professional fields such as farming, construction, gardeners, movers, etc. Their ability to transport large quantities of goods, equipment and material makes them invaluable in these industries.

According to the Health and Safety Executive (HSE) UK [1], nearly a quarter of all deaths involving workplace transport are related to reversing operations. Brenac and Fournier [2] investigated accidents between pedestrians and reversing vehicles in France and found that commercial and cargo vehicles are overrepresented in these collisions. Even if an accident does not cause injury, there is at least costly damage to vehicles, equipment, and premises. To our knowledge, there is no statistical number that indicates the cases in which trailers are involved. However, reverse operation with a trailer is a dangerous task, which gets increasingly complex with increasing length and bulk.

Modern vehicles have built-in features, such as path-lined backup cameras for easier reversal. Affordable aftermarket kits exist, but they are tedious to install and mainly fit standard vehicles. Although beneficial for trailers, such systems are costly or unavailable, especially for older models. Designed for standard vehicles, these systems do not accommodate custom vehicle-trailer dynamics requiring different sensors and inputs.

We provide a solution to the aforementioned problems at an approximate price tag of about EUR 150. Our main proposal is the concept of a backup camera system with rear and side views for trailers. Lines resembling the path of the trailer are calculated on the basis of inverse kinematics, using real-time measurements captured by individual sensors, such as the steering angle and the angle between the vehicle and the trailer. The cameras not only cover areas not visible to the driver when reversing but also provide imagery to display dynamic reversing lines to the driver in an augmented view. Our contributions can be summarized as follows.

- A camera setup for a trailer, providing a bird's perspective onto the area around and behind the trailer;
- a sensor setup based on cost-efficient and low-energy components for the steering and the hinge angles;
- path prediction based on inverse kinematics, using real-time data from said sensors; and
- a mobile interface to provide a real-time view with backup line functions to communicate the travel path to the driver.

The remainder of the paper is organized as follows. Section II reviews related work. Section III is dedicated to a description of the concepts and algorithms involved, while Section IV describes our prototypical implementation. In Section V, we describe an exemplary implementation and some experimental results, while we conclude with a discussion of our findings in Section VI.

II. Related Work

Related work is grouped into four different areas: (i) image stitching and multi-camera calibration, including perspective projection; (ii) technology used to measure the steering angle, which is also used in autonomous driving; (iii) trailer-related developments, which are scarse, yet consider the measurement of the hitch angle; and finally (iv) inverse kinematics, an area primarily considered in robotics. Note that we abstain from an in-depth discussion of markets and autonomous driving as its own research area.

A. Multi-Camera Systems

In the context of vehicle assistance systems, bird's eye views have been studied by Liu *et al.* [3], who present a surround view system with six cameras. Like other equivalent systems, they try to reduce blind spots. Li and Hai [4] present an approach using fisheye cameras, proposing easy calibration of a multi-camera fisheye system. Lee *et al.* [5] present a bird's eye system in-painting missing areas (*i.e.,* the actual area occupied by the vehicle) from previous images. Dueholm *et al.* [6] combine bird's eye views and object detection to perform multiperspective vehicle tracking on highways.

Bijo *et al.* [7] describe a system using embedded hardware, studying the efforts to map images and to provide a transformed and stitched view. Zhang *et al.* [8] present a real-time capable surround-view camera system implemented on a Digital Signal Processor (DSP). Geometry alignment and view synthesis are applied to each frame to produce an output video at 30 FPS. Pan *et al.* [9] further propose a DSP-driven setup to stitch panoramic images from cameras on vehicle mirrors.

In our work, we mainly follow the standard literature [10], [11] to facilitate image stitching and perspective warping, however, for a trailer system. For simplicity, we ignore particular aspects related to the optimization of image quality or strict camera placement.

B. Steering Angle Measurements

Modern cars usually measure the steering angle through embedded hardware and provide it to higher-level control units on the Controller Area Network Bus (CAN-Bus) [12]. In the literature, several methods for measuring the steering angle are mentioned, ranging from Computer Vision (CV) based approaches to additional hardware.

Dev *et al.* [13] demonstrate a method to estimate the steering angle based on lane markings recorded through a camera. Saleem *et al.* [14] provides a broad overview of steering angle estimation methods based on imagery, involving Convolutional Neural Networks (CNNs). Hiligsmann and Riendeau [15] introduce a programmable monolithic CMOS linear Hall effect sensor. The sensor is dedicated to high-precision contact-free rotary position detection. Paul and Kim [16] demonstrate a contact-free device mounted on the steering column using an optocoupler for angular encoding. Zacharia *et al.* [17] make use of gear wheels with magnetic sensors mounted on the steering column. Moussa *et al.* [18] proposes the use of smartphones in combination with Extended Kalman Filters (EKF) to estimate the steering angle, mounting the mobile device rigidly to the steering wheel.

Our steering angle solution was inspired by the work of Zacharia *et al.* [17], due to the ease of implementation in the era of 3D printers and programmable microcontrollers.

C. Hitch Angle Measurements

Altafini *et al.* [19] introduce a control scheme for trailer stabilization while driving backward with a miniaturized truck. Kong and Kosko [20] developed a traditional neural network to estimate the path of a trailer. Nakamura *et al.* [21] describe a kinematic model using the Japanese Yamamiya trailer model in forward and backward motion. De Saxe *et al.* [22] use a camera and template matching to estimate homographies. Fuchs *et al.* [23] propose a similar system using markers.

In general, sensors and encoders promise improved robustness. In this respect Rolfes [24] proposes a cost-efficient trailer backup system, which integrates directly with vehicles of a specific brand (Ford) and existing electronic control units (ECUs). Zhu *et al.* [25] measure the hitch angle of a trailer with a rear-mounted single-line LiDAR scanner. Bahramgiri *et al.* [26] use a set of radar sensors to estimate the hitch angle. Lou *et al.* [27] also use existing LiDAR sensors of an autonomous truck.Our idea is similar to the one of Zhu *et al.* [25], however, we propose the use of ultrasonic sensors similar to Bahramgiri *et al.* [26], and a different assembly.

D. Inverse Kinematics

Both Kinematics and Inverse Kinematics are big fields of research with many applications in robotics and other domains [28]. For example, Klug *et al.* [29] compute configurations of total stations, while Parger *et al.* [30] employ inverse kinematics in a similar manner to compute joint positions of human arms. Aristidou *et al.* [31] provide an overview of the use of inverse kinematics in Computer Graphics (CG) and Ruuskanen [32] discusses its use for game character animation.

In the context of trailers, Pradalier and Usher [33] present a method for stabilizing trailers by focusing on the hitch angle rather than using the steering angle to guide the trajectory. Kinematic models were also used by Beglini *et al.* [34] or Leng *et al.* [35] to study or prevent the so-called *Jackknife phenomenon* during advance driving (*i.e.,* the trailer pushing the towing vehicle and eventually hitting it).

We do not consider the *Jackknife phenomenon* in our work, as we consider it's occurance to be problematic in cases where a vehicle is driving forwards rapidly, rather than very slowly backwards. We use both angles, *i.e.,* the steering angle and the hitch angle, to feed our inverse kinematics simulation for trajectory prediction. However, the calculation of the trajectory itself is inspired by the work of Pradalier and Usher [33].

III. System and Algorithmic Concept

Our approach can be divided into three main groups: **video processing** (green), **kinematic model** (orange), and **user interface** (black), according to Figure 1. Note that we consider **physical properties** (gray) as a given list of custom-made constant configuration parameters for a vehicle-trailer combination.

a) Video Processing.: A number of cameras are mounted in a downward position around the trailer, to increase visibility and reduce blind spots. The streams are gathered via cable in a central processing unit, the fish-eye distortion is removed, and the streams undergo an orthographic projection to reassemble a bird's eye view. In an offline step, all cameras are calibrated and registered to share a common world origin. The final result is a combined multi-camera bird's eye view video stream.

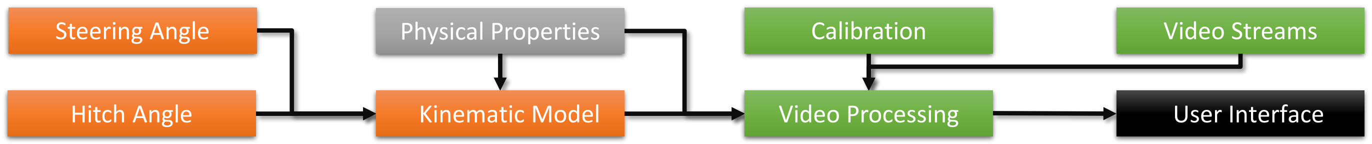


Fig. 1. System components: **(Orange)** depicts input nodes necessary to compute the guidance lines. **(Gray)** refers to physical properties such as vehicle geometry and image properties. **(Green)** refers to the video processing part, where streams together with their calibration (intrinsic and extrinsic parameters) are stitched into a live-stream shown on the user interface **(Black)**.

b) Kinematic Model.: The trailer path prediction is based on the steering angle of the towing vehicle, the hitch angle, and the vehicle-trailer geometry. We use inverse kinematics to model this dynamic pendulum-like system and feed it with the described inputs. For a certain configuration, we can predict the future position of the trailer and manifest it as lines for driver guidance by passing them on for visualization in the user interface.

c) User Interface.: The user interface gathers all the information from the previous steps to create an augmented view for the user. The guidance lines based on the estimated trajectory are superimposed on the video imagery, taking into account the calibration and trailer position.

A. Video Processing

To create a bird's eye perspective, an offline calibration must first be performed for each camera to determine the intrinsic parameters. The extrinsics are estimated through pairwise registration. Using both, we can undistort the images and project them orthographically according to their spatial configuration. The final bird's eye view is generated by blending all orthographic images into one, properly considering overlapping areas. A graphical description of these steps is shown in Figure 2.

a) Camera Calibration: The calibration is based on a well-known approach by Kannala and Brandt [36], detecting multiple correspondences between image points (*i.e.,* camera pixel locations) and real-world coordinates (*i.e.,* object points) on a defined calibration pattern (see top right of Figure 2). ChArUco boards combine the commonly used chessboard pattern with ArUco markers and are therefore advantageous, as they combine easily detectable markers and good corner detection of regular chessboard patterns. As a result, correspondences can be created even when parts of the whole board are not clearly visible or not in view overall.

The image points are refined to subpixel accuracy, while the corresponding object points are the 3D real-world coordinates of checkerboard corners with respective x and y values, while the z value remains 0 for all points due to the planarity of the pattern. These 2D-3D correspondences from multiple images are used to estimate the camera intrinsics and distortion parameters. The intrinsics of the camera and the distortion parameters are used to create undistorted images (see top left of Figure 2) with pixels identified by (u,v), which serve as input for further steps in the pipeline.

b) Registration: A registration step is now performed to unify the individual coordinate systems of all cameras into one global coordinate frame. This merely involves the definition of a global origin, and estimating the transformation *i.e.,* the 6 degree-of-freedom (DOF) poses, between all cameras.

Usually, no more than two cameras observe the same physical space. Since cameras therefore form a chain-like structure, placing the global origin somewhere in the center of this chain (*i.e.,* the middle camera of a setup with an uneven number of cameras) is advantageous for numerical reasons. Hence, placing a regular marker properly, it is observed in pairs of cameras and, respectively, their undistorted images. The marker establishes common 3D coordinates, *i.e.,* metric coordinates, observed in both images.

A camera pose $P = [R|T] \in \mathbb{R}^{3\times 4}$ represents the pose of a single camera with respect to the marker. Following Fleck *et al.* [37], we can calculate a pairwise registration and align two cameras by calculating

$$P_{ji} = P_i \cdot P_j^{-1} \tag{1}$$

where i, j are the camera indices. Iterating over all pairs of cameras with overlapping views gives an overall registration of all cameras with respect to each other.

A byproduct of having a marker placed flat on the ground is the estimation of the ground plane, which is crucial to establish the orthographic projection and to introduce additional metric information, if desired, into the system.

c) Orthographic Projection: From the undistorted images, an orthographic projection is established [10] [11]. Using the known registration information between cameras, the individual parts of a final common orthographic frame are filled out, as shown in the middle left of Figure 2. Each pixel (u',v') within a desired orthographic image is mapped backward to a pixel in the respective undistorted camera image (u,v). This results in a set of orthographic projections that can be stitched without further translation.

This mapping is intentionally backward to depend only on the dimensions of the desired output image. A forward mapping would result in visual artifacts as also discussed by Wagner *et al.* [38] for panoramic surfaces. Undistorted images are mapped only once to orthographic images for efficient stitching later on without any additional effort.

d) Image blending: To finally create a stitched view as seen in Figure 2 on the lower right, a weight mask W_i for each camera i is used. This mask matches the output image size, with values from 0 to 1, initially set to 0.

We identify overlapping regions by generating a binary mask M_i, where each pixel is 1 if it corresponds to a projected pixel from camera i, and 0 otherwise. The overlap regions between adjacent cameras i, j are identified by intersecting the masks $\Delta M_{i,j} = M_i \wedge M_j$ (*i.e.,* a bit-wise AND of the masks).

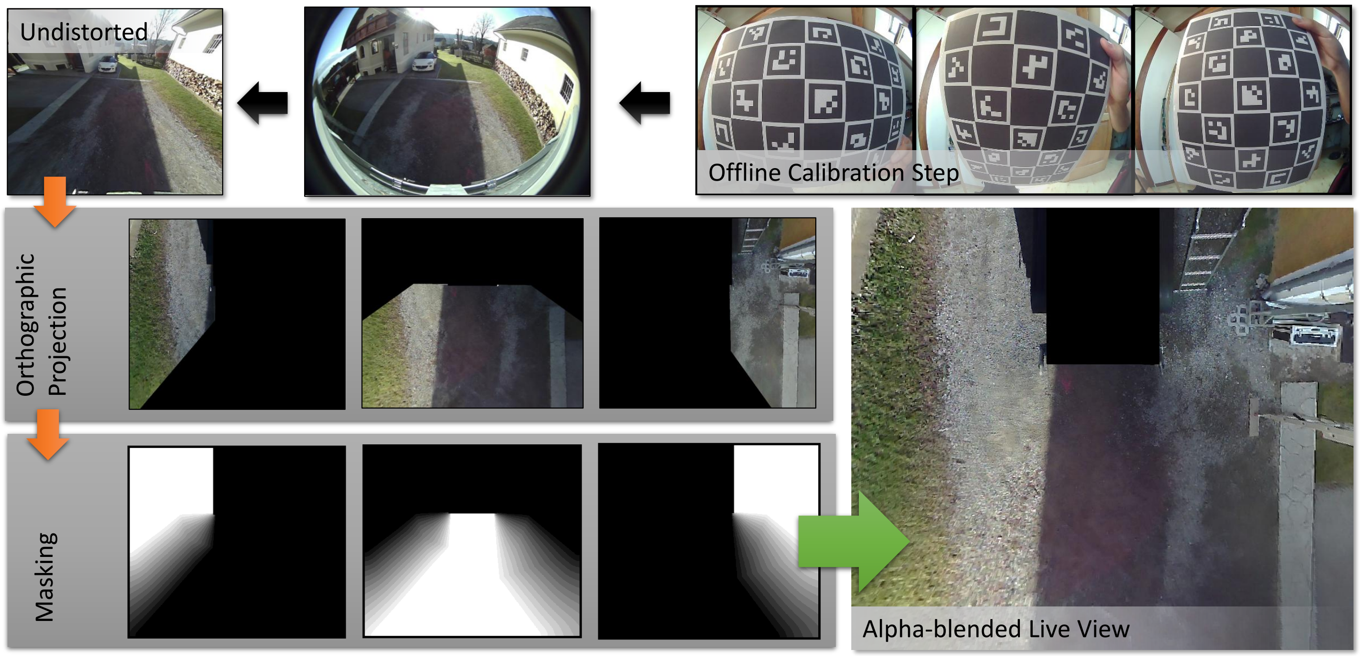


Fig. 2. Overview of the individual parts of the system. Based on calibration images (**Offline Calibration Step**), the calibration parameters are calculated in an offline step once for each camera. The undistorted images (**Undistorted**) are then orthographically projected based on the registration information calculated (**Orthographic Projection**). Masks for proper alpha blending (**Masking**) are used to combine all orthographic views into a common bird's eye view (**Alpha-blended Live View**).

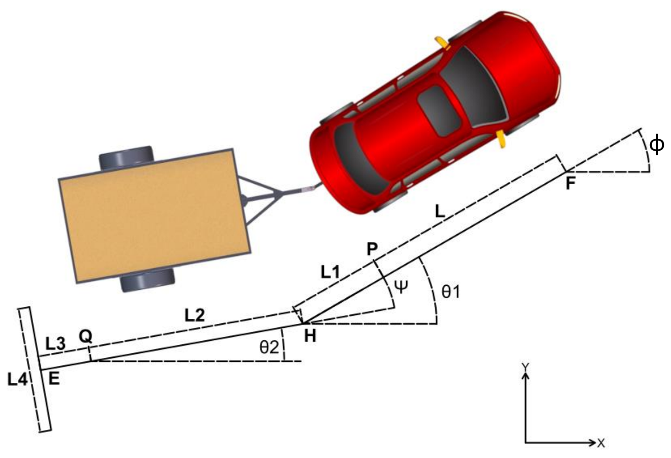


Fig. 3. Visualization of a vehicle trailer combination and the corresponding model with parameters required for our guidance line simulation. A list of variables is given in Table I.

F	front vehicle axle	P	rear vehicle axle
Q	trailer axle	H	trailer coupling
E	rear trailer end		
L	distance F to P	L1	distance P to H
L2	distance H to Q	L3	distance Q to E
L4	trailer width		
ϕ	*steering angle*	Ψ	*hitch angle*
$\theta 1$	auxiliary angle	$\theta 2$	auxiliary angle

TABLE I
PARAMETERS USED IN OUR KINEMATICS MODEL, AS SHOWN IN FIGURE 3.

To isolate the unique regions of each camera, we XOR the overlapping mask $\Delta M_{i,j}$ with M_i, respectively M_j, resulting in U_i and U_j.

The output weight mask W_i is initially set to U_i, as all unique pixels are excluded from further weighting calculations. For the overlapping regions identified as $\Delta M_{i,j}$, weights are determined based on their relative distances to the contours of the unique masks, *i.e.,* U_i and U_j. After extracting the outer contours (polygons) of the masks U_i and U_j, the pixel weights within the overlapping region $\Delta M_{i,j}$ are calculated based on the distances. Let d_i and d_j be the respective distances for a single pixel (u,v), the weight for the respective pixel in W_i and W_j is given by

$$w_i(u,v) = \frac{d_i}{d_i + d_j}, \qquad w_j(u,v) = \frac{d_j}{d_i + d_j} = 1 - w_i(u,v). \quad (2)$$

In other words, the areas with overlap are filled in W in a weighted fashion, as seen in Figure 2 on the lower left. The masks W thus resemble alpha-blending masks to stitch all images in the final bird's eye view.

B. Kinematic Model

The prediction of the trailer trajectory strongly depends on the overall vehicle-trailer geometry, the *steering angle* ϕ and the *hitch angle* Ψ. We aim to adapt a kinematic model for a vehicle-trailer combination to calculate the position of a certain point p at any time. More specifically, we want to compute the positions of points located on the rear of the trailer. Therefore, we need a set of parameters for the vehicle-trailer combination as shown in Figure 3, respectively Table I.

In standard car kinematics equations, the distance between the front and rear axles of the car, denoted as L, is required. This distance is measured between the point F and the central reference point P. In particular, L is the only value documented in the car's approval certificate, meaning that all other measurements must be determined manually. The length from the rear axle to the trailer coupling H is represented as L_1. For the trailer, the only measurement needed is L_2, which represents the distance between the trailer coupling and the center of the trailer axle Q.

In addition to these measurements, the vehicle's steering angle ϕ and the hitch angle Ψ, both defined relative to the vehicle's heading, are required. Using this information, the system is described by the following equations:

$$\dot{x} = v\cos(\theta_1), \quad \dot{y} = v\sin(\theta_1), \quad \dot{\theta}_1 = \frac{\tan(\phi)}{L}, \\ \dot{\Psi} = -v\frac{L_1\sin(\phi)\cos(\Psi) + L_2\sin(\phi) + L\cos(\phi)\sin(\Psi)}{L_2 L\cos(\phi)}. \tag{3}$$

The next position of P, respectively it's x and y components, θ_1, and Ψ can be estimated by adding the computed change multiplied by the desired time difference to the previous values. In Eqn. (3), v denotes a linear velocity of point P.

The kinematic model proposed by Pradalier and Usher [33] focuses on the starting and end location of the point P. In our case, this point is not relevant, and we focus only on the end of the trailer, specifically the newly introduced point E. At the start of the kinematic simulation, the trailer rear must be parallel to the x-axis, and the coordinate center is located at point E. We first need to calculate the position of the point H, which is located at (0, $L_2 + L_3$). By splitting the length L_1 into its respective x and y components and using the angle Ψ, we can then calculate the position of point P as follows:

$$P = (H_x + L_1\cos(\Psi), \quad H_y + L_1\sin(\Psi)). \tag{4}$$

With this initial starting point and the angles, which remain constant during the simulation, the computation process can be initiated. This setup provides the necessary basis for iteratively calculating the motion of the system.

The simulation proceeds either for a fixed number of iterations or until the hitch angle exceeds a predefined threshold, indicating that the trailer is pivoting sharply and may potentially collide with the vehicle. Until one of these conditions is met, the next values for x, y, θ_1, and Ψ are computed using Euler's method and the previously mentioned Eqn. (3). This method involves calculating the tangent at the current position P and taking a small step in its direction with a specified velocity v. The simulation yields a list of positions for point P along with the corresponding values for Ψ.

To finally arrive at the two corners of the trailer, we use the estimated P based on Eqn. (4) and incorporate the estimate of Ψ, but additionally scale the output points by a factor s. Connecting the resulting points results in the expected trajectory of the corner points. In Figure 4 the initial kinematics of a vehicle-trailer combination is shown, together with some exemplary steps and the estimated trajectory.

IV. Prototypical Implementation

Certain requirements have to be met in order to implement the described system. We will briefly discuss our approach to building a versatile system that fits most potential vehicle-trailer combinations.

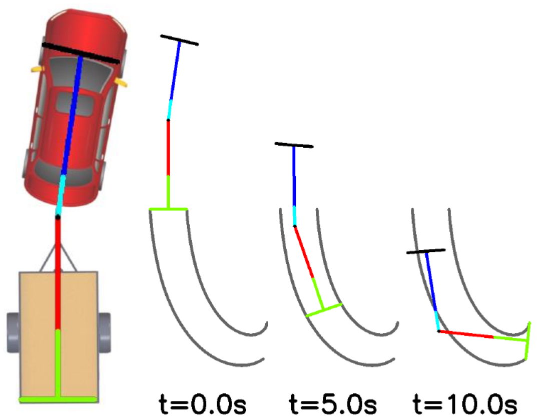


Fig. 4. Visualization of the kinematic model during the guidance line calculation with the angles $\phi = 5°$ and $\Psi = -8°$. The simulation was performed with a velocity of 1 m/s over a duration of 10 seconds.

A. Camera Setup

Although wireless cameras could be used instead, in our implementation, we use standard 2-megapixel USB cameras with fisheye lenses, directly wired to the central processing unit to reduce latency. The three cameras used shared enough overlap to allow marker-based 3D registration. The used ArUco marker (1.2×1.2 m) was placed on a flat surface and detected in each camera using the ArUco library (OpenCV), as presented by Garrido-Jurado *et al.* [39].

Note that the calculation of camera intrinsics, the overall registration, the proper undistortion of cameras, respectively, the orthographic mapping, and the blending masks have to be done only once. Storing the mappings in lookup tables, imagery can thereby be efficiently translated into a bird's eye view at runtime.

B. Steering Angle

In our prototypical implementation, the steering angle measurement is performed using a custom-made board based on an *ESP8266 Wemos D1 Mini* paired with a *KY-040* rotary encoder. The encoder is coupled to the steering rod using a 1:2 step-up gear. The encoder operates at a voltage of up to 5V and encodes a full 360° rotation into 30 pulses. It is connected to the ESP8266 via its `5V`, `GND`, `CLK`, and `DT` pins. To debounce the internal switches for the `CLK` and `DT` signals, two additional 100nF capacitors are added. This ensures stable signal processing by the microcontroller.

For each pulse, the encoder's `CLK` pin changes its logic level state, which can be processed using an interrupt service routine (ISR) on the ESP8266. To determine the direction of rotation, the `DT` pin of the encoder is also monitored. This pin changes state with each rotation step, but with a slight phase offset from the `CLK` signal. If the `DT` pin value differs from the `CLK` pin, the encoder turns clockwise; if the values match, the encoder turns counterclockwise. The realization of the described sensor is shown in Figure 5. Note that the data are wirelessly transmitted to a given computing unit.

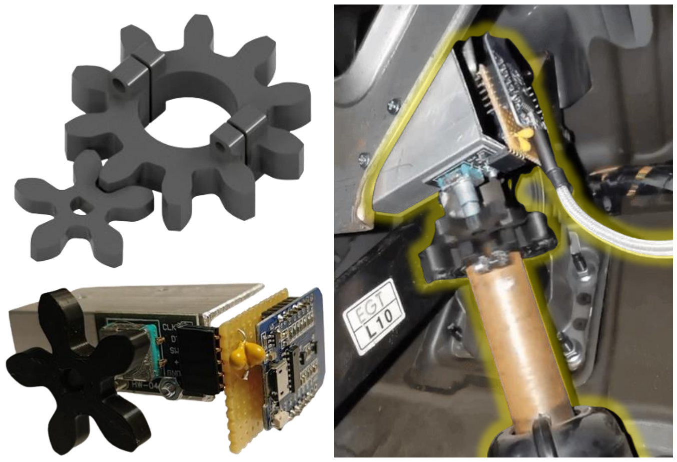


Fig. 5. A custom-built steering angle sensor (highlighted in yellow) using a 2:1 gear ratio system and a rotary encoder to measure the steering angle. An ESP-based SoC is used to transmit the data wirelessly.

C. Hitch Angle

The hitch angle denotes the angle between the trailer and the vehicle. Focusing on off-the-shelf hardware, and independent and cost-efficient solutions, we developed an easy-to-adapt setup that can be further improved or replaced.

Our implementation uses an ESP32 development board with two *HC-SR04* ultrasonic sensors that provide a measurable range of 2cm to 4m and can achieve an accuracy of up to 3mm under ideal conditions. A level shifter is used to decrease the 5V from the HC-SR04 to 3.3V, which matches the logic level of the ESP32.

The distance measurement process works by first pulling the `TRIG` pin to 5V for a duration of 10µs. This action triggers the sensor to emit eight sonic bursts at a frequency of 40kHz. The distance is then encoded as a pulse sent out by the sensor on the ECHO pin, which represents the travel time of one burst. This pulse duration, t, is converted to meters with $l(t) = \frac{340 \cdot t}{2}$.

Factors such as angled surfaces, environmental influences, and signal reflections can impact its accuracy. To improve measurement reliability, a median filter was applied to five sequential measurements, which were then averaged with the previous median value. This process improves the accuracy to approximately $\pm$ 0.5cm, although some limitations remain.

An exemplary assembly is shown in Figure 6. Note that in order to save computational power on the microcontroller, only the sampling and median/mean calculations are executed on the ESP32 for the measurements a and b and trasmitted wirelessly. The final angle calculation is given by:

$$\Psi = \tan^{-1}(1 - a/b). \tag{5}$$

D. Kinematic Simulation and User Interface

Our kinematic simulation is executed on a central processing unit. The velocity is set to $v = -1$ m/s, as the trailer is reversing, and a step size of 1cm is chosen. This means that a total distance of, for example, 1m is covered in 100 iterations. The central processing unit also receives the steer and hitch angles from the respective sensors.

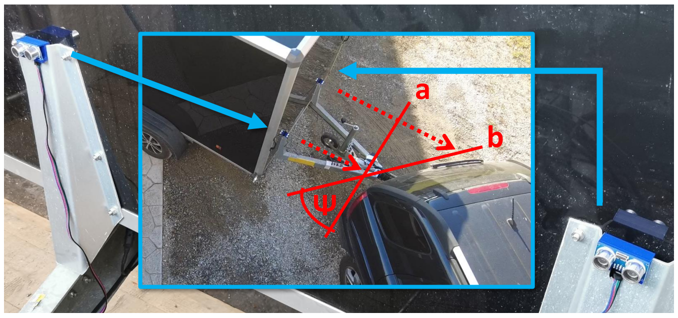


Fig. 6. Hitch angle estimation with two ultrasonic distance sensors mounted on each side of the trailer.

The user interface combines the bird's eye view and the computed trajectory to provide guidance lines in a video stream. Using physical properties we can align the guidance lines with the corners of the trailer in the combined image, to produce a convincing end result.

A web server is used to serve the stream over the network to client devices. This approach allows for the visualization of the user interface over the built-in browser of the car's multimedia system, besides an external device. Note that while not strictly coherent with various scientific definitions of user interfaces, we denote this part as such, although we do not have any active input modalities for customization, which could easily be added in future work.

V. Experimental Results

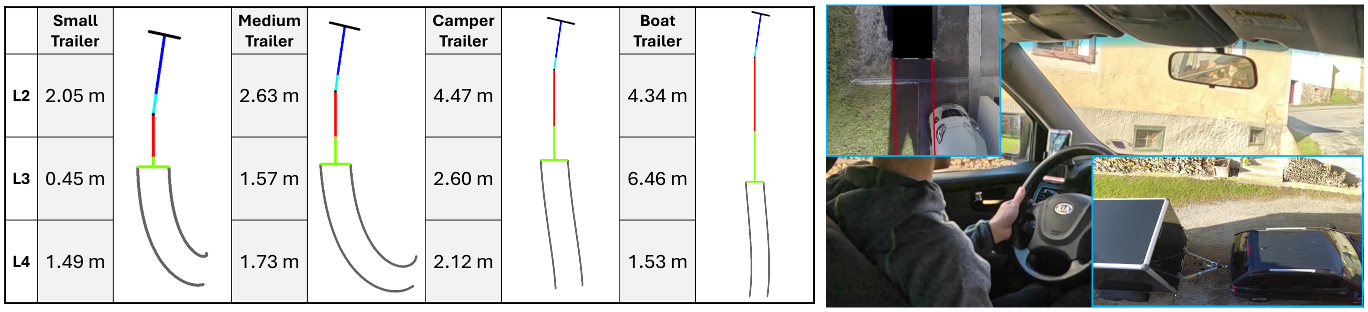
We perform a number of different experiments and reversing maneuvers to examine our system. In all our tests, we drove backwards, recording the measurements and the respective user interface.Although an embedded system can certainly be used as the central processing unit, we used a *HP Spectre x360* mobile computer with an *i7-8550U Intel* processor and 8 GB of RAM with *Ubuntu* 24.04. The central processing unit integrates the kinematic simulation and the measurements and provides the user interface to mobile clients. Note that we did not perform any optimization on the algorithms or investigate computational bottlenecks.

We used a *KIA Sportage 2009* to tow, respectively push, a *Variant 2005C3* trailer. Note that this towing vehicle does not provide the steer angle over the ODB interface.

A. Power Consumption and Timing

We measured the power consumption of each of our sensor assemblies and timed the individual algorithmic, yet unoptimized, processing steps. The hitch angle sensor used 150mA at 5V during operation and 80mA at 5V in its idle state, whereas the steer angle sensor used 80mA at 5V during operation and during idling. The output image size was chosen to be 780$\times$890 pixels, covering approximately 3m to either side of the trailer and 3.8m to the back.

The time for the undistortion/orthographic mapping takes about 16ms, while the time for blending the bird's eye view takes an additional 5ms. The time to measure the two angles and compute these lines takes on average 45ms, while the actual drawing takes around 11ms.



	Small Trailer		Medium Trailer		Camper Trailer		Boat Trailer	
L2	2.05 m		2.63 m		4.47 m		4.34 m	
L3	0.45 m		1.57 m		2.60 m		6.46 m	
L4	1.49 m		1.73 m		2.12 m		1.53 m	

Fig. 7. (Left) Final position of simulated trailers using the same steering (5°) and hitch (8°) angle inputs. (Right) Reversing straight into a parking spot, with an overlaid view of the assisted parking visualization and an external view of the vehicle trailer combination.

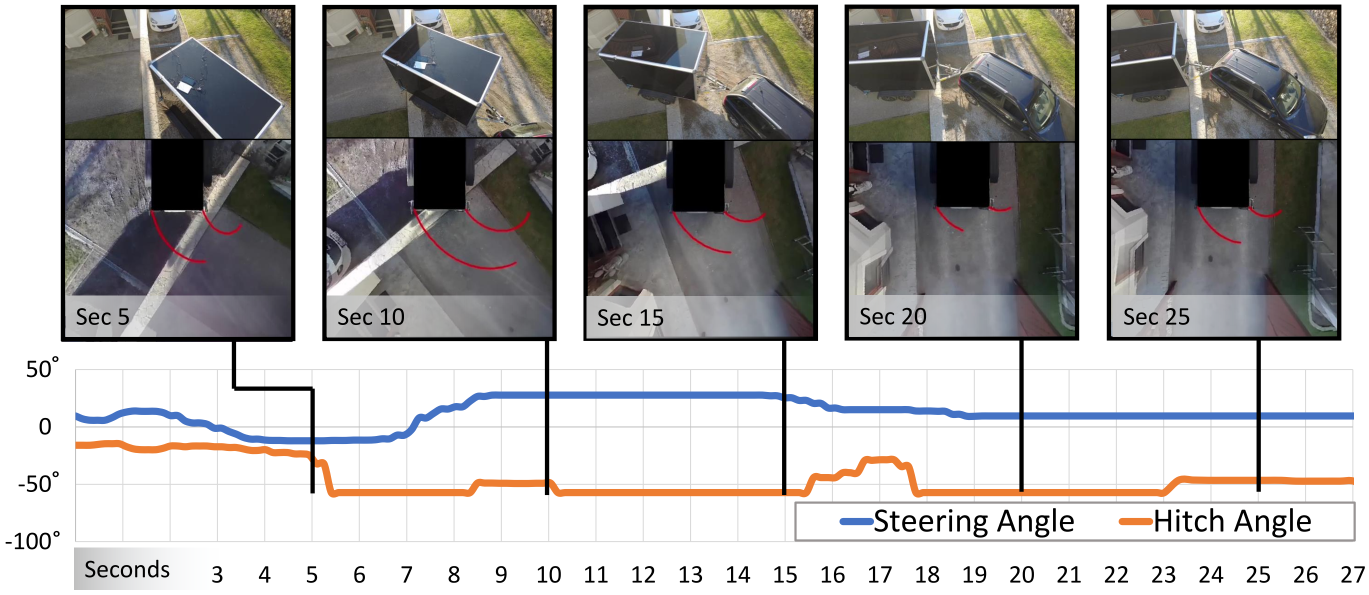


Fig. 8. Angular behavior and path predictions while backing into a parking spot. While the steer angle measurements accurately reflect the actual steering action, the hitch angle suffers from occasional errors, while still being sufficiently accurate to plausibly predict the trailer trajectory.

B. Straight Backwards and Cornering

During straight driving, the steer angle is accurately measured at 0°. Despite a stable hitch angle, due to the error in the distance measurements, the hitch angle fluctuates between ± 2 degrees. This fluctuation causes the guiding lines to move even though there is no real angle change between the car and the trailer. Our straight parking scenario is shown in Figure 7.

Our system eases parking by predicting the trailer's position based on steering, enabling quick adjustments. The bird's eye view helps estimate remaining space in the parking spot. We conducted several tests; however, in all the scenarios we evaluated, *i.e.,* garage, parking spot, and gravel road, we obtained similar results. We observe that the guidance lines become highly beneficial when maneuvering into small openings or tight spaces. However, some artifacts are visible in the overlapping regions, *e.g.,* a ghosting shadow of the white car in Figure 7, top left. These artifacts result from the projection of tall objects onto the ground surface by two adjacent cameras, observing the object from different angles.

Figure 8 shows a cornering drive into a parking spot over a time period of 27 seconds. The steering angle works counterintuitively to the hitch angle and the guidance lines are updated accordingly.

C. Different Vehicle-Trailer Combinations in Simulation

To demonstrate our inverse kinematics model capabilities, we perform four simulations with different trailers and let them revert with similar steering and hitch angle inputs.

Figure 7 shows the different final positions ordered by trailer size. Consequently, our model takes the changed parameters into account and produces a meaningful output. For our experiments, we did not have access to any other real trailer to verify the output in the real world; however, based on the behavior we observed with our trailer, the inverse kinematics model seems reasonable enough for real-world applications.

VI. Discussion and Concluding Remarks

In this work, we introduce a system for assisted trailer parking using a camera system and a kinematic model. During the testing of our system, we encountered several limitations in the actual implementation. For brevity, only a selected number will be explained in this section; however, for those, ideas are provided for future improvements.

The hitch angle sensor system incorporates a pair of ultrasonic sensors, chosen for their widespread availability and affordability. However, these sensors have significant distance errors and lack waterproofing. To enhance performance, exploring sensor alternatives like LiDAR or Time-of-Flight (ToF) sensors might be beneficial, because of their superior accuracy and reliability under challenging conditions.

The steering angle sensor was tailored specifically for the vehicle in use. Future efforts might aim at creating a more adaptable solution that can be customized in various ways. The mobile computer should also be replaced with a single-board-computer (SBC) to enable the processing unit to permanently mount on the vehicle or trailer, respectively. The video processing pipeline can be improved to deal with brightness changes across cameras and to decrease any latencies.

Although trailer use is common and the actual problem of trailer parking is well known, performing a user study following scientifically profound standards is very difficult and time-consuming in practice. The authors are aware of the complexity and, while a user study could provide valuable insights, without imminent interest expressed by industrial partners, it is not planned for the time being.

Acknowledgment

Funded by the European Union under Grant Agreement No. 101092861. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Health and Safety Executive (HSE) UK, "Reversing," https://www.hse.gov.uk/workplacetransport/information/reversing.htm, 2024, online; accessed Dec. 18, 2024.

[2] T. Brenac and J.-Y. Fournier, "Collisions between pedestrians and reversing vehicles in public settings in france," *Open Transportation Journal*, vol. 12, pp. 33–42, 2018.

[3] Y.-C. Liu, K.-Y. Lin, and Y.-S. Chen, "Bird's-eye view vision system for vehicle surrounding monitoring," in *Robot Vision*, 2008, pp. 207–218.

[4] S. Li and Y. Hai, "Easy calibration of a blind-spot-free fisheye camera system using a scene of a parking space," *IEEE Trans. Intelligent Transportation Systems*, vol. 12, no. 1, pp. 232–242, 2011.

[5] J. Lee, M. Kim, S. Lee, and S. Hwang, "Real-time downward view generation of a vehicle using around view monitor system," *IEEE Trans. Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3447–3456, 2020.

[6] J. V. Dueholm, M. S. Kristoffersen, R. K. Satzoda, T. B. Moeslund, and M. M. Trivedi, "Trajectories and maneuvers of surrounding vehicles with panoramic camera arrays," *IEEE Trans. on Intelligent Vehicles*, vol. 1, no. 2, pp. 203–214, 2016.

[7] B. Thomas, R. Chithambaran, Y. Picard, and C. Cougnard, "Development of a cost effective bird's eye view parking assistance system," in *IEEE Recent Adv. in Intelligent Comp. Systems*, 2011, pp. 461–466.

[8] B. Zhang, V. Appia, I. Pekkucuksen, A. Umit Batur, P. Shastry, S. Liu, S. Sivasankaran, and K. Chitnis, "A Surround View Camera Solution for Embedded Systems," in *CVPR Workshops*, 2014.

[9] J. Pan, V. Appia, J. Villarreal, L. Weaver, and D.-K. Kwon, "Rear-stitched view panorama: A low-power embedded implementation for smart rear-view mirrors on vehicles," in *CVPR Workshops*, 2017, pp. 1184–1193.

[10] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[11] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003.

[12] W. Voss, *A Comprehensible Guide to Controller Area Network*. Copperhill Technologies Corporation, 2008.

[13] V. S. Dev, V. V. Sajith Variyar, and K. P. Soman, "Steering angle estimation for autonomous vehicle," in *IEEE Int. Conference on Advances in Computing*, 2017.

[14] H. Saleem, F. Riaz, L. Mostarda, M. A. Niazi, A. Rafiq, and S. Saeed, "Steering angle prediction techniques for autonomous ground vehicles: A review," *IEEE Access*, vol. 9, pp. 78 567–78 585, 2021.

[15] V. Hiligsmann and P. Riendeau, "Monolithic 360 degrees rotary position sensor IC," in *IEEE Sensors*, 2004.

[16] D. Paul and T. H. Kim, "On the feasibility of the optical steering wheel sensor: Modeling and control," *International Journal of Automotive Technology*, vol. 12, no. 5, pp. 661–669, 2011.

[17] S. Zacharia, T. George, and E. Rufus, "Implementation of Steering Wheel Angle Sensor System with Controlled Area Network," in *Int. Conf. on Intelligent Comp., Instrumentation and Control Techn.*, 2017.

[18] M. Moussa, A. Moussa, and N. El-Sheimy, "Steering Angle Assisted Vehicular Navigation Using Portable Devices in GNSS-Denied Environments," *MDPI Sensors*, vol. 19, no. 7, p. 1618, 2019.

[19] A. Claudio, A. Speranzon, and B. Wahlberg, "A Feedback Control Scheme for Reversing a Truck and Trailer Vehicle," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 6, pp. 915–922, 2002.

[20] S. G. Kong and B. Kosko, "Adaptive fuzzy systems for backing up a truck-and-trailer," *IEEE Trans. on Neural Networks*, vol. 3 2, pp. 211–23, 1992.

[21] Y. Nakamura, H. Ezaki, Y. Tan, and W. Chung, "Design of steering mechanism and control of nonholonomic trailer systems," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 3, pp. 367–374, 2001.

[22] C. De Saxe and D. Cebon, "A visual template-matching method for articulation angle measurement," in *IEEE Int. Conf. on Intelligent Transportation*, 2015.

[23] C. Fuchs, F. Neuhaus, and D. Paulus, "Advanced 3-D Trailer Pose Estimation for Articulated Vehicles," in *IEEE Symp. on Intelligent Vehicles*, 2015.

[24] N. Rolfes, "Requirement Modeling of Pro Trailer Backup Assist [tm]," *SAE Int. Journal of Passenger Cars*, vol. 10, no. 1, pp. 116–126, 2017.

[25] H. Zhu, W. Yuan, C. Wang, and M. Yang, "Reverse Path Following Method of a Tractor-Trailer Vehicle in Real-World Implementation," in *IEEE Int. Conference on Robotics and Biomimetics*, 2022.

[26] M. Bahramgiri, S. Nooshabadi, K. T. Olutomilayo, and D. R. Fuhrmann, "Automotive radar-based hitch angle tracking technique for trailer backup assistant systems," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1922–1933, 2023.

[27] W. Luo, C. Jiang, Q. Zhang, Z. Pan, and L. Heng, "Lidar-Assisted Hitch Angle Estimation System for Self-Driving Truck," in *IEEE Symp. on Intelligent Vehicle*, 2024.

[28] J. Craig, *Introduction to Robotics*, 4th ed. Pearson Deutschland, 2021. [Online]. Available: https://elibrary.pearson.de/book/99.150005/9781292164953

[29] C. Klug, D. Schmalstieg, T. Gloor, and C. Arth, "A complete workflow for automatic forward kinematics model extraction of robotic total stations using the denavit-hartenberg convention," *Journal of Intelligent & Robotic Systems*, vol. 95, pp. 311–329, 2019.

[30] M. Parger, J. H. Mueller, D. Schmalstieg, and M. Steinberger, "Human upper-body inverse kinematics for increased embodiment in consumer-grade virtual reality," in *VRST*, 2018.

[31] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: A survey," in *Computer graphics forum*, vol. 37 (6). Wiley Online Library, 2018, pp. 35–58.

[32] A. Ruuskanen, "Inverse kinematics in game character animation," Bachelor's Thesis, Kajaanin Ammattikorkeakoulu University of Applied Sciences, 2018.

[33] C. Pradalier and K. Usher, "A simple and efficient control scheme to reverse a tractor-trailer system on a trajectory," in *ICRA*, 2007.

[34] M. Beglini, L. Lanari, and G. Oriolo, "Anti-jackknifing control of tractor-trailer vehicles via intrinsically stable mpc," in *ICRA*, 2020, pp. 8806–8812.

[35] Z. Leng, Y. Wang, M. Xin, and M. A. Minor, "The effect of sideslip on jackknife limits during low speed trailer operation," *Robotics*, vol. 11, no. 6, p. 133, 2022.

[36] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *PAMI*, vol. 28, no. 8, pp. 1335–1340, 2006.

[37] P. Fleck, C. Arth, C. Pirchheim, and D. Schmalstieg, "Tracking and mapping with a swarm of heterogeneous clients," in *ISMAR*, 2015.

[38] D. Wagner, A. Mulloni, T. Langlotz, and D. Schmalstieg, "Real-Time Panoramic Mapping and Tracking on Mobile Phones," in *IEEE VR*, 2010.

[39] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.