Emergency Corridor Building on Multi-Lane Motorways with Autonomous Model Cars

Jurij Kuzmic, Günter Rudolph and Fabian Ostermann
Department of Computer Science, TU Dortmund University, Otto-Hahn-Str. 14, Dortmund, Germany

Keywords: Emergency Corridor Building, Model Cars, Standing Traffic, Lane Detection, Filtered Canny Edge Detector, Autonomous Driving, Vehicle-to-Vehicle (V2V) Communication, Autonomous Traffic.

Abstract: This paper introduces an algorithm for forming an emergency corridor on motorways with autonomous vehicles. This algorithm can be used in slow-moving traffic and in standing traffic scenarios. In addition, several autonomous model vehicles were assembled for the experiments in this work in order to test this algorithm in real-world use. Furthermore, a model motorway was constructed that resembles a real three-lane motorway. The Filtered Canny Edge Detector algorithm, also previously published by us, is used to recognise the lanes of the motorway from camera images. Also, in this work, this lane detector is further extended and improved for use with model vehicles in the real environment. Our experiments also show successful emergency lane formations for four different events. Finally, possible future work in this area is presented.

1 INTRODUCTION

Unforeseeable accidents happen again and again on roads and motorways. Sometimes serious accidents can occur on the motorways as vehicles are travelling at high speeds. This leads to traffic jams and road closures. In those cases, most countries have regulations for forming an *emergency corridor* (or rescue lane) for emergency vehicles arriving at the scene of an accident. But human drivers sometimes forget to form such an emergency lane for police and rescue vehicles when in a traffic jam. In extreme cases, where the traffic comes to a hold in a chaotic constellation due to a sudden accident, it is no longer a straightforward task to form the rescue corridor. When the vehicles come to a standstill, they are usually too close to each other. In addition, there is the problem that even if a rescue lane can be formed after the vehicles have stopped and one emergency vehicle drives past, the drivers of the standing vehicles close the rescue lane again. This repeatedly leads to the obstruction of other police or rescue vehicles.

The problem is not new and has been discussed for some time (Dębiński, Jukowski and Bohatkiewicz, 2018). Currently, there is no optimal solution. Some countries impose heavy fines for obstructing emergency vehicles (e.g., Austrian Federal Ministry of Mobility, 2023). To counteract this problem, an Emergency Vehicle Warning System has already been introduced, which uses radio communication to warn other vehicles about an emergency vehicle (Buchenscheit, Schaub, Kargl and Weber, 2010). To completely eliminate this problem, the emergency corridor must be formed automatically by the autonomous vehicles that probably soon are likely to be approved for regular road traffic. The question is not *if*, but *when* autonomous vehicles will come onto our roads.

According to our research, at the time of the development of our algorithm, there was only the work we published in the field of automatic formation of an emergency corridor with autonomous vehicles (Kuzmic and Rudolph, 2020). In addition, there were only three patents for warning and communication systems for the formation of the emergency lane through vehicle-to-vehicle communication (Colella and Herman, 2018; Marin, 1998; Siegel, 2003). This allows for sharing information between road users, for example about accident location or the route of an approaching emergency vehicle.

To reduce the required budget for our study to a minimum, we are using model vehicles. Of course, the use of real cars would be preferable, but since we would need at least a dozen of them to simulate a rescue corridor situation, this is not achievable even for highly equipped research labs. With our approach,

Kuzmic, J., Rudolph, G. and Ostermann, F.
Emergency Corridor Building on Multi-Lane Motorways with Autonomous Model Cars.
DOI: 10.5220/0012731700003705
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 9th International Conference on Internet of Things, Big Data and Security (IoTBDS 2024)*, pages 321-328
ISBN: 978-989-758-699-6; ISSN: 2184-4976
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

scaling future experiments to having tens or hundreds of cars is in a reasonable scope. In contrast to a computer simulation, the model cars experience problems of real world scenarios, for example, sensor noise, inaccuracies in motor control, or blurred images obtained from a camera in motion.

2 ALGORITHM FOR EMERGENCY CORRIDOR BUILDING

Algorithm 1 is developed for the formation of the emergency corridor in *standing traffic* (when the traffic has come to a standstill). The algorithm works with four states. The normal state is called *NORMAL_FORWARD*. This state's logic was already published in (Kuzmic and Rudolph, 2020) and handles the slow-moving traffic scenario. Vehicles usually are driving at high speeds on the motorway. If the speed falls below a certain value (e.g. 30 km/h) because of heavy traffic, the vehicles begin to form a rescue corridor as they drive. This means that the vehicles on the last lane (the leftmost lane, as we assume right-hand traffic) move to the left lane marking and the vehicles on the adjacent lane move to the right lane marking. The vehicles in the other lanes do not change their lane position. Counting of the lanes starts in the direction of driving from right to left, as the first lane is decided to be the lane one drives onto first. Please note, that our algorithm also works for left-hand traffic.

For unforeseeable reasons, it can happen that vehicles come to a sudden standstill and human drivers forget to form an emergency corridor. The rescue lane for police and rescue vehicles is then missing. With our algorithm, the drivers of these vehicles can attempt to safely form an emergency corridor out of such a chaotic situation at the push of a button, or in the case of fully automated vehicles, the cars decide to do this themselves.

Algorithm 1, requires the generation of a local map for each individual vehicle. This means that all vehicles share information about their positions, including their current lane number. By sending the information of their own identification number (in practice, this can be the licence plate number), their own lane position and the identification number of the vehicle in front, it is possible to calculate the order of the vehicles on all the lanes and to create a local map. Then, the vehicles can decide which vehicle has to start with forming the emergency corridor. Three further states are used to form the corridor from standing traffic. They allow to form the rescue lane systematically (row by row). The vehicles closest to the vehicles involved in the accident will start. As soon as a vehicle has completed its task, it notifies the vehicle behind it on the same lane. The corresponding states *WAIT*, *EXEC_BACKWARD*, and *EXEC_FORWARD* will be explained in more detail in Section 4.

Algorithm 1: Building an Emergency Corridor: Standing Traffic

```
Input  : StartSpeed = 0, ActualFrontDistance > 0,
         MinFrontDistance = 3, ActualLane ≤ MaxLanes,
         MaxLanes > 1, State ≥ 0
Output: Control of the vehicle
send message LOCALMAP ;
receive messages POSITION ;
create a local map of the vehicles and their front vehicles ;
if ActualLane = MaxLanes or ActualLane = MaxLanes − 1 then
    if first vehicle in the lane then
        State = EXEC_BACKWARD ;
    else
        State = WAIT ;
for every single step do
    switch State do
        case WAIT do
            do not move the vehicle ;
            wait for message CHANGESTATE to change the state ;
        case EXEC_BACKWARD do
            if ActualFrontDistance < MinFrontDistance then
                move the vehicle straight backwards ;
                check the rear distance ;
            set State = EXEC_FORWARD ;
        case EXEC_FORWARD do
            reduce the safety distance to the vehicle in front ;
            increase the steering behaviour ;
            move the vehicle forwards ;
            execute Algorithm for Building an Emergency Corridor:
              Slow-Moving Traffic ;
            if Execution completed then
                send CHANGESTATE to the rear vehicle with
                  State = EXEC_BACKWARD ;
        case NORMAL_FORWARD do
            increase the safety distance to the vehicle in front ;
            reduce the steering behaviour ;
            move the vehicle forwards ;
            execute Algorithm for Building an Emergency Corridor:
              Slow-Moving Traffic ;
        /* No valid ID of the vehicle or accident vehicle   */
        otherwise do
            do not move the vehicle ;
```

Algorithm 1: Formation of an emergency corridor in a standing traffic.

3 AUTONOMOUS MODEL CARS

The SunFounder PiCar-S Smart Car kit for the Raspberry Pi (SunFounder, 2022) was used to assemble the model vehicles from the model making area. This already contains several motors, wheels, control plates and sensors for the construction (Fig. 1, left). The cost of each individual kit was around 90 Euros at the time of development (the Raspberry Pi not included). Our own measurements show that the model vehicles reach speeds of 0.61 m/s (2.22 km/h).

We have adapted and expanded the SunFounder PiCar-S Smart Car kit for our use case. Figure 1 shows a comparison between the original model vehicle and our modified version.

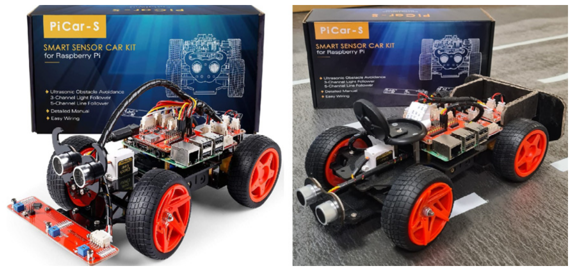
Figure 1: SunFounder PiCar-S Smart Car kit for Raspberry Pi. Left: Original. Right: Modified.

3.1 Construction of the Model Cars

The body of the model vehicles consists of several die-cut acrylic panels, which are fixed in place with a few screws. This results in slight inaccuracies in the steering of the model vehicles. A stepper motor is responsible for steering the model vehicles. But, the steering can only be carried out in steps of 2° instead of 1°: If the stepper motor is moved in 1° steps, the front wheels do not move. This is due to inaccuracies in the motor control design.

The next step was to add a Raspberry Pi 3 B (Kofler, Kühnast and Scherbeck, 2021) and a Raspberry Pi V2 camera module (RaspberryPi, 2016) to the kit. The camera module allows to optically detect the lane markings. The camera was installed in a suitable camera housing. This housing was attached with two spacers and fixed at a certain height from the motorway at a certain angle to the motorway. The ultrasonic sensor for distance measurement was installed lower so as not to restrict the camera's view.

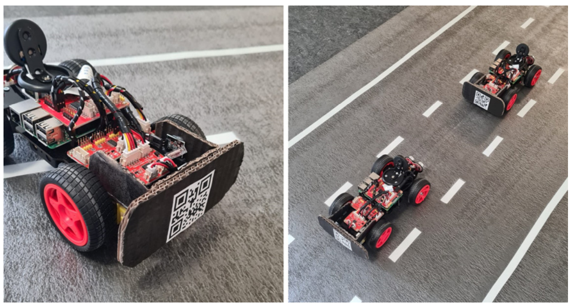
Figure 2: Extensions of the SunFounder PiCar-S Smart Car kit for Raspberry Pi. Left: QR-Code bumper. Right: Distance measurement with ultrasonic sensor.

A black cardboard bumper was attached to the back of each model vehicle so that the ultrasonic sensors find a straight surface for measuring the distance to the model vehicle in front (Fig. 2). The affixed QR code provides the information of the vehicle's identification number (ID).

3.2 Construction of the Model Motorway

A miniature model motorway was set up to carry out the experiments. This allowed to test the previously developed algorithm in different situations. In addition, this test track could be used to test the improved lane detection with the model vehicles and to adapt it for this purpose (details follow in Section 3.3). The test motorway was also adapted to the size of the model vehicles used. The test track resembles a three-lane motorway (Fig. 3, left). To carry out the planned experiments also on a two-lane motorway, one of the lanes can be "closed". Eight metres of impact sound insulation for laminate was used for the ground of the track. This material resembles the colouring of a real road surface. White insulating tape was used to apply the lane markings. Figure 3 (right) shows the test model motorway including the nine modified model vehicles that were available for our experiments.

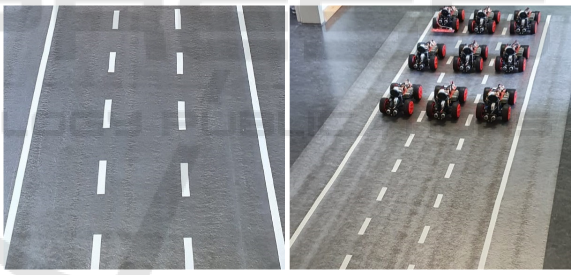
Figure 3: Construction of the three-lane model motorway. Left: Motorway without model cars. Right: Motorway with model cars.

3.3 Lane Detection

For lane detection with the model vehicles, the previously published Filtered Canny Edge Detector algorithm (Kuzmic and Rudolph, 2021) was used. It is further extended and optimized for the model vehicles in the present paper. All that is written in bold font is a novel addition, that substantially improves lane detection on real camera images.

0. As a preliminary work, the camera should be aligned and calibrated. Only the vital areas should be visible in the image (only the road). E.g., the bonnet of the car, low-

hanging power lines, etc., should not be visible.

1. Create a greyscale image from the coloured input image.
2. Create a Canny edge image from the grayscale image (Canny, 1986).
3. Apply probabilistic Hough transform (Stephens, 1990) to the edge image (2D coordinates for P and Q).
4. Calculate the gradient m for each straight line.
5. Calculate the angle with atan(m) from the gradient m.
6. Ignore all lines with an angle θ below α=**25**° and above β = 80° (α < θ < β).
7. Calculate the points of intersection with the X-axis IX for the remaining lines.
8. **Sort left and right lines from the centre of the image B into separate data fields.**
9. **Ignore left lines with a negative gradient m<0 and right lines with a positive gradient m>0**.
10. **Save intersection points SX, angle θ, gradient m and P and Q coordinates of the lines for left and right lines respectively**.
11. **Calculate intersection points of left and right lines. Accept them in a predefined area between line intersection point 1 and line intersection point 2**.
12. **Origin of these lines must be in the lower two-thirds of the image**.
13. From the centre of the image B, find the smallest distance D = |IX-B| left and right.
14. Search in a predefined radius R for further intersection points IX left and right.
15. **Lines found in the predefined radius R must contain a similar angle to the X-axis. Angle threshold W with $|\theta_1 - \theta_2| < W$.**
16. Calculate an average function (straight line) left and right from the found lines.
17. Calculate the centre of the track from these two functions left and right.

The runtime measurements were carried out on nine different autonomous model vehicles. By implementing and executing the Filtered Canny Edge Detector algorithm with multithreading on the Raspberry Pi 3 B, we achieved slightly below 30 frames per second (FPS) as shown in Table 1. Also, Table 1 shows that the model vehicle with ID 9 achieves only 24.5 FPS. This is due to an individual hardware defect of this particular model car.

Table 1: Runtime of lane detection on the Raspberry Pi 3 B.

Model Car [ID]	Lane [No]	Runtime [FPS]
1	3	29.69
2	2	29.33
3	1	29.24
4	3	29.46
5	2	28.88
6	1	28.00
7	3	29.44
8	2	29.14
9	1	24.50

Figure 4 shows the improved Filtered Canny Edge Detector in use on real-world camera images. As the lighting conditions in the real environment can change, automatic adjustment of the threshold values for the Canny Edge Detector is required. For this reason, the two threshold values for the filtered Canny algorithm are dynamically adapted. The median value of the greyscale image is used for this normalisation (Rosebrock, 2015).

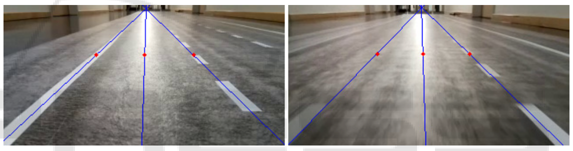
Figure 4: Filtered Canny Edge Detector as used by the model cars on the model motorway. Left: Perfect lighting conditions. Right: Blurred vision because of car movement.

3.4 Distance Detection and Steering Behaviour

For the model vehicle, the distance measurement to the model vehicle in front is realised by an ultrasonic sensor. The horizontal detection range of this ultrasonic sensor is about 15°. This detection range is sufficient for forming the emergency corridor, as the vehicles autonomously keep a sufficient safety distance and the rescue lane can therefore be formed without extreme steering. Most importance has the precision of the distance sensor. The ultrasonic sensor in use sometimes detects false input pulses from the surroundings, which lead to incorrect distance measurement. This was also reported by ELEC Freaks (2022).

During the development of the steering behaviour of the vehicles, various tests revealed that the actual pixel difference in the lane detection can be used to control the steering. The pixel difference from the centre of the lane to the centre of the model vehicle can be used directly as the steering step in degrees.

Only a maximum steering threshold needs to be set. We chose a maximum value of 10°.

3.5 Communication Between Self-Driving Model Cars

Communication between the model vehicles is established via wireless LAN (Wi-Fi). At this point, it is the easiest way to realise communication, as the model vehicles are already connected to Wi-Fi for remote development. This simplifies the execution of the various scripts on the Raspberry Pi and also makes troubleshooting easier thanks to the Secure Shell (SSH) network protocol. Also, the output of the executed scripts can be monitored. Also, SSH access to the model vehicle via Wi-Fi makes it easy to calibrate the stepper motor and to configure the model vehicles.

Communication between the model vehicles allows them to react autonomously to various messages. For example, as soon as an accident or collision involving the model vehicle is recognised, the vehicle performs emergency braking and notifies the vehicles in the vicinity. The message contains a name as well as information about the accident vehicle's identification number. The various messages of our specification are listed in Table 2.

Table 2: Overview of all possible messages including the description for communication between autonomous model cars.

Name	Description
COLLISION	Collision/rotation of the vehicle
FORWARD	Forwarding the incoming message
RESCUE	Notifying emergency vehicles of an accident
LANE	Own lane of the vehicle
TOTALLANES	Total lanes of the motorway
LOCALMAP	Building the local map
POSITION	Own lane of the vehicle and ID of the vehicle in front
STOP	Stop own vehicle
CHANGESTATE	Change of state

As already explained in Section 2, communication between the model vehicles can be used to create the required local map of the vehicles on the model motorway to form an emergency corridor in the event of a standing traffic. The *LOCALMAP* message is sent to request the information from each individual vehicle. The vehicles that receive this message respond with the *POSITION* message. This message contains information about the vehicle's own lane and the identification number of the model vehicle in front in the same lane. Each individual model vehicle can then independently create its own local map with all vehicles that are present in the real environment. The *CHANGESTATE* message is sent to change the state for executing the formation of the emergency corridor (cf. Algorithm 1).

4 EXPERIMENTS

As before, we used our virtual simulator (Kuzmic and Rudolph, 2020) to perform preliminary testing of our new Algorithm 1. To proof that the results transfer to the real world, we now attempt to carry out experiments with our real model vehicles. This allows the behaviour of the algorithm for forming the emergency corridor in slow-moving and standing traffic to be checked in reality. The different scenarios are explained below using the illustrations. Videos are also available online for all scenarios (Kuzmic, 2023).

4.1 Building an Emergency Corridor: Slow-Moving Traffic

This experiment shows the algorithm for forming the emergency corridor in slow-moving traffic with the autonomous model vehicles, which are in the state *NORMAL_FOWARD*. The emergency lane is formed between the leftmost and the adjacent lane. When forming the rescue lane, the position of the left or right front wheel of the model vehicle must be compared with the previously recognised lane markings. At this point, the previously introduced lane detection (Filtered Canny Edge Detector) from Section 3.3 is used. This means that each model vehicle knows in which direction (left or right) it has to move in order to form a rescue corridor depend on its own lane. As Figure 5, left, shows, the rescue lane can be successfully formed by the autonomous model vehicles.

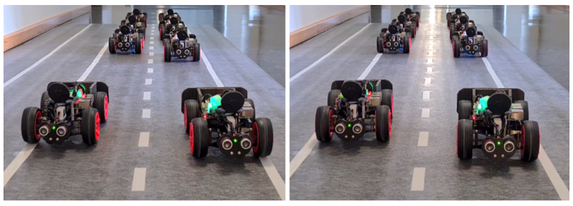
Figure 5: Building an emergency corridor at slow-moving traffic on a two-lane motorway. Left: Emergency corridor about to open. Right: Emergency corridor opened and will close if cars can speed up again.

In a slow-moving traffic, the model vehicles automatically drive to the left or right to open a corridor for emergency vehicles. If the speed of the model vehicles increases, the opened emergency corridor is automatically closed again (Fig. 5, right). This ensures that the vehicles have formed a rescue lane for the police and emergency vehicles at all times after braking.

4.2 Building an Emergency Corridor: Obstacles in Front

In this section, the formation of the emergency corridor in the event of various obstacles on a motorway is tested. If the speed decreases, it is theoretically irrelevant for the model vehicles for what reason it is. The position of an obstructive object is more important than its type (e.g. crashed car, crossing animal, etc.). As soon as an object is detected in the vehicle's own lane and is approaching, the speed of the model vehicle is reduced. Then, the emergency corridor is opened automatically when the speed falls below a certain value (e.g. 30 km/h). As Figure 6 shows, the emergency corridor can be successfully formed with the model vehicles on a two-lane model motorway in the event of an accident.

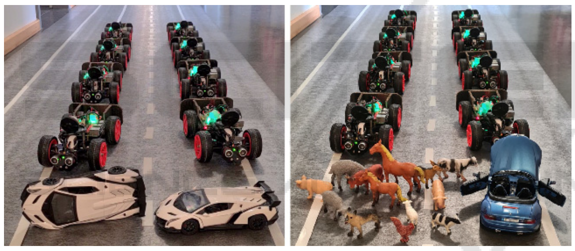
Figure 6: Successfully established emergency corridor in the event of an accident on a two-lane motorway. Left: Car accident. Right: Animals crossing.

The rescue corridor can also be successfully formed on a three-lane motorway in the event of an accident with a closed lane. This is shown in Figure 7.

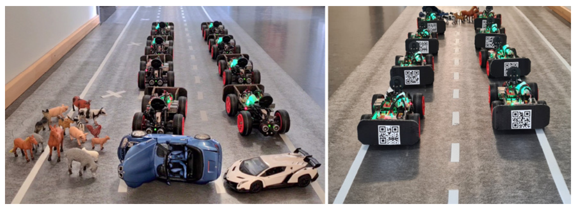
Figure 7: Successfully established emergency corridor in the event of an accident on a three-lane motorway with a closed lane. Left: Front-view. Right: Rear-view.

Additionally, the formation of the emergency corridor with autonomous model vehicles can also be carried out on a three-lane motorway. This is shown in Figure 8. In the previously published simulator, some experiments were also successfully carried out on a four-lane motorway with different lane widths (Kuzmic and Rudolph, 2020).

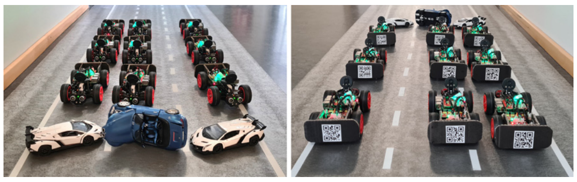
Figure 8: Successfully established emergency corridor in the event of an accident on a three-lane motorway. Left: Front-view. Right: Rear-view.

4.3 Building an Emergency Corridor: Front Tire Bursting

To test whether the emergency corridor can be formed at any random time with the autonomous model vehicles, the front tyre of the model vehicle was simulated to burst. In this experiment, the impact sensor was used to detect an unintended and sudden movement of the model vehicle that we caused by hand. The occurrence of this movement was unpredictable for the model. Thus, the crashed vehicle recognised its accident and performed an emergency braking manoeuvre. As shown in Figure 9, left, this model vehicle is in the third lane (most right lane in the picture). Once the accident and the unforeseeable turn has been recognised, all vehicles in the vicinity are notified in order to avoid collisions. These following model vehicles can then immediately react to the incoming message and independently decide how to form the emergency corridor.

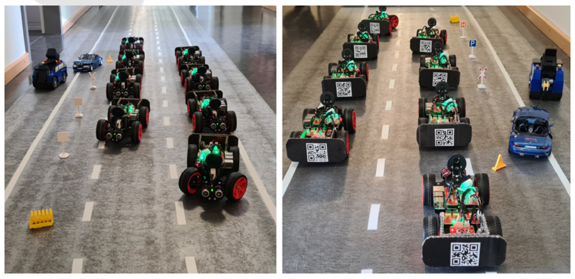
Figure 9: Model realisation of a front tire bursting while driving. Left: Front-view. Right: Rear-view.

The first model vehicle in the second lane also had to perform an emergency braking, as this model vehicle is close to the vehicle involved having the accident. However, this first model vehicle in the second lane was still able to form the emergency corridor already as it steered safely during the emergency braking. All other vehicles maintained the safety distance and were therefore further away. This allowed these model vehicles to brake normally and at the same time open the rescue corridor for the police and rescue vehicles (Fig. 9, right). The fact that such a complex manoeuvre (simultaneous braking and targeted steering) can be carried out is a decisive advantage of autonomous vehicles, as it could never be performed reliably by a human in such fractions of a second.

4.4 Building an Emergency Corridor: Standing Traffic

This section presents the experimental tests of Algorithm 1 for the formation of the emergency corridor in a standing traffic in our real model environment. For this, it is also necessary for each individual model vehicle to generate a local map through communication and optical evaluation of the model vehicle in front. By exchanging its own identification number, its own lane position and the identification number of the vehicle in front, the sequence of the model vehicles in the various lanes can be created from the local map.

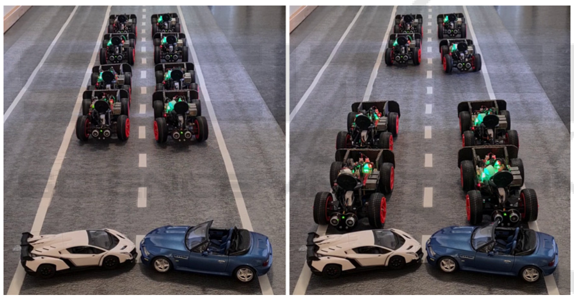
Figure 10: Building an emergency corridor in the event of an accident in a standing traffic on a two-lane motorway. Left: First row execution. Right: Third row execution.

This means that each model vehicle can make the decision for itself which vehicles should start to form the rescue corridor. The model vehicles systematically form the rescue lane for the police and rescue vehicles row by row. At this point, the formation of the rescue lane in standing traffic takes place via four states (see Algorithm 1). The model vehicles closest to the accident vehicles start forming the rescue lane (Fig. 10, left). If the distance is too short for the emergency lane to be formed, the model vehicles automatically drive backwards (*EXEC_BACKWARD*). As our autonomous model vehicles do not have rear sensors for distance measurement, the vehicles behind notify the vehicles in front of them of the distance. As soon as the model vehicle in front has completed the formation of the emergency corridor, it notifies the model vehicle behind it in the same lane. From the third row onwards, the distance for a rescue lane is sufficient so that these model vehicles no longer have to drive backwards (*EXEC_FORWARD)*. This can be seen in Figure 10, right. From the sixth row onwards, the model vehicles are in the *NORMAL_FORWARD* state and can build the emergency corridor in a slow-moving traffic. As Figure 11 shows, the rescue lane can be successfully formed on a two-lane highway in a standing traffic.

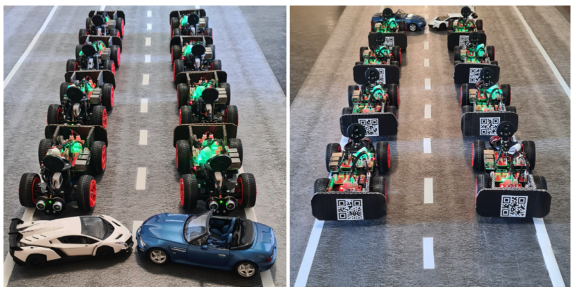
Figure 11: Successfully established emergency corridor in the event of an accident in a standing traffic on a two-lane motorway. Left: Front-view. Right: Rear-view.

Also, the emergency corridor can be successfully formed automatically on a three-lane highway. This is shown in Figure 12.

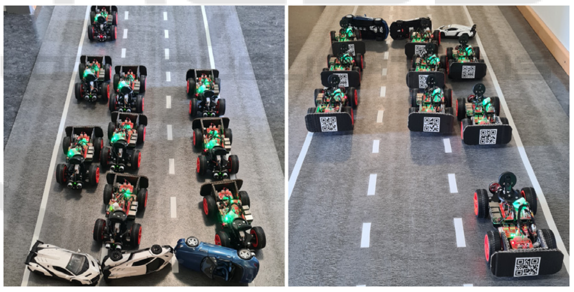
Figure 12: Successfully established emergency corridor in the event of an accident in a standing traffic on a three-lane motorway. Left: Front-view. Right: Rear-view.

5 CONCLUSIONS

The algorithm developed for emergency lane formation was successfully tested in reality using the adapted lane detection (Filtered Canny Edge Detector). The correct functionality of the algorithm was demonstrated experimentally for slow-moving and standing traffic in a real-world situation using model cars. The model cars acted completely autonomously. The reason and the external circumstances for the formation of the emergency corridor are irrelevant. This could be, for example, slow-moving traffic, animal crossings, a burst front tyre, a car collision or any other unforeseeable event. Conducting the experiments with model vehicles showed more difficulties than our previously published simulation (Kuzmic and Rudolph, 2020). In a real environment, the quality and measurement tolerance of the sensors and the design of the model vehicles for the test play an important role. The previously published and in this paper extended and improved lane detection (Filtered Canny Edge Detector) could additionally be further optimised for hardware with limited resources (Raspberry Pi 3 B). However, about 30 FPS, which we achieved by CPU parallelization, already enabled us to run our lane detection in real-time.

In addition, selected examples of the experiments in this work were published online (Kuzmic, 2023) with references to our scientific publications.

For future work, the car body of the model vehicles could be improved. It is also recommended to use a better ultrasonic sensor for measuring the distance to the vehicle in front. Finally, it would be most interesting to test how the algorithm presented in this paper performs in the real world with real autonomous vehicles in real size on a real motorway.

REFERENCES

Austrian Federal Ministry of Mobility, 2023. *Rettungsgasse*. Oesterreich.gv.at. [online]. Available at: https://www.oesterreich.gv.at/themen/mobilitaet/kfz/10/Seite.063130/Seite.065000.html. Accessed: 20.10.2023.

Buchenscheit, A., Schaub, F., Kargl, F., Weber, M., 2018. *A VANET-based emergency vehicle warning system*, 2009 IEEE Vehicular Networking Conference (VNC), pp. 1-8, ISBN 978-1-4244-5685-7.

Canny, J., 1986. *A Computational Approach to Edge Detection*. In Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence, Bd. PAMI-8. 6, pp. 679-698, DOI: 10.1109/TPAMI.1986.4767851.

Colella, N., Herman, D. A., 2018. *Emergency corridor utilizing vehicle-to-vehicle communication*. US-Pat. US20170352268A1. Ford Global Technologies LLC.

ELEC Freaks, 2022. *Ultrasonic Ranging Module HC - SR04*. Cdn.sparkfun.com. [online]. Available at: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf. Accessed: 17.02.2024.

Dębiński, M., Jukowski, M., Bohatkiewicz, J., 2018. *Emergency Corridors - Necessary Solutions in a Modern Road Network*, MATEC Web of Conferences 231 (GAMBIT 2018), pp. 1-6.

Kofler, M., Kühnast, C., Scherbeck, C., 2021. *Raspberry Pi: Das umfassende Handbuch*. Galileo Press, pp. 17-44, ISBN 978-3-8362-8353-3.

Kuzmic, J., 2023. *Building an Emergency Corridor with Autonomous Cars*. YouTube.com. [online]. Available at: https://www.youtube.com/watch?v=OEgJETouExc&list=PLIzKtWM9Wfqen3TxLNnE-noxu980jd_gy&index=14. Accessed: 17.02.2024.

Kuzmic, J., Rudolph, G., 2020. *Unity 3D Simulator of Autonomous Motorway Traffic Applied to Emergency Corridor Building*. In Proceedings of the 5th International Conference on Internet of Things, Big Data and Security (IoTBDS), pp. 197-204, ISBN 978-989-758-426-8.

Kuzmic, J., Rudolph, G., 2021. *Comparison between Filtered Canny Edge Detector and Convolutional Neural Network for Real Time Lane Detection in a Unity 3D Simulator*. In Proceedings of the 6th International Conference on Internet of Things, Big Data and Security (IoTBDS), pp. 148-155, ISBN 978-989-758-504-3.

Marin, R. T., 1998. *Emergency vehicle proximity warning and communication system*. US-Pat. US4794394A.

Stephens, R. S., 1990. *A probabilistic approach to the Hough Transform*. In Proceedings of the British Machine Vision Conference, British Machine Vision Association, pp. 12.1-12.6, DOI:10.5244/c.4.12.

RaspberryPi, 2016. *Raspberry Pi Camera Module 2*. Raspberrypi.com. [online]. Available at: https://www.raspberrypi.com/products/camera-module-v2/. Accessed: 15.12.2022.

Rosebrock, A., 2015. *Zero-parameter, automatic Canny edge detection with Python and OpenCV*. Pyimagesearch.com. [online]. Available at: https://pyimagesearch.com/2015/04/06/zeroparameter-automatic-canny-edge-detection-with-python-and-opencv/. Accessed: 13.04.2023.

Siegel, M. A., 2003. *Emergency vehicle alert system*. US-Pat. US20030043056A1.

SunFounder, 2022. *Raspberry Pi Smart Car Kit (Picar-S) for Intermediate*. Sunfounder.com. [online]. Available at: https://docs.sunfounder.com/projects/picar-s/en/stable/introduction.html. Accessed: 28.11.2022.