

UDK 004.932

**B. Żygadło**

## AUTOMATIC PARALLEL CAR PARKING

**Abstract.** This paper presents a method of automatic parallel car parking. The first part describes the principles of vehicle movements. Trajectories of front and back axles are presented, as well as the difference between turning of a car forward and backwards. The second part describes the developed algorithm. Position of the vehicle is determined using readouts of six proximity sensors and mathematical calculations. Two kinds of parking are considered. The first one is a maneuver in which the vehicle moves only backwards on the reverse gear. The trajectory of the car and its mathematical description is provided. The second maneuver allows the car multiple change of the direction of movement and is applied when there is not enough of space for performing the first one. The starting position of the vehicle is determined by simulating the reverse maneuver, that is to say, going out of the parking spot. Amount of space required for parking is calculated differently for each maneuver. The last part of the paper consist of a description of 3-D animation of the system.

**Keywords:** Parking, automatization, 3-D graphics

**Б. Жигадло**

## АВТОМАТИЧЕСКАЯ ПАРАЛЛЕЛЬНАЯ ПАРКОВКА АВТОМОБИЛЯ

**Аннотация.** Представлен метод автоматической параллельной парковки автомобиля. Первая часть описывает принципы движения автомобиля. Вторая представляет спроектированный алгоритм. В статье рассмотрены два вида парковки. Первым является маневр, в котором автомобиль двигается только задним ходом. Вторым разрешает автомобилю менять направление движения многократно и применяется в случае, когда нет достаточно места для выполнения первого маневра. Последняя часть статьи описывает 3-D анимацию системы парковки.

**Ключевые слова:** парковка, автоматизация, 3-D графика

**Б. Жигадло**

## АВТОМАТИЧНЕ ПАРАЛЛЕЛЬНЕ ПАРКУВАННЯ АВТОМОБІЛЯ

**Анотація.** Представлен метод автоматичного паркування автомобіля. Перша частина описує принципи руху автомобіля. Друга представляє спроектиований алгоритм. У статті розглядається два види паркування. Першим є маневр, у якому автомобіль рухається тільки заднім ходом. Другий дозволяє автомобілю змінювати напрямок руху багаторазово і використовується тоді, коли немає достатнього місця для виконання першого маневру. Остання частина статті описує 3-D анімацію системи паркування.

**Ключові слова:** паркування, автоматизація, 3-D графіка

### 1. Introduction

Automatic parking is a system that moves automatically a vehicle from a street into a parking spot [1]. Its aim is to enhance the comfort and safety of driving. Parking is achieved by means of coordinated control of the steering angle and speed which takes into account the actual situation in the environment to ensure collision-free motion within the available space [2]. There are three distinguished types of parking: parallel, perpendicular and angle. The most difficult one – parallel parking [3 – 4], has been presented in this paper.

### 2. Vehicle movements principles

A turning vehicle rotates along a circle. The position of the center of the circle depends on the angle of the steering wheels [5].

Actually moving along a straight line may be also considered as circular motion, in which the radius of the curve is infinite.

Fig. 1, a present's movement of a car while turning forward. All points of the car rotate around the same point – the rotation center. However, they move along different curves that are they have different radius of rotation.

The curves of movement of all four wheels have been shown. The radius of the turn of the wheels is perpendicular to its elevation plan [6]. Since two back wheels are not steered, they are always parallel to each other and the radius of their rotation lies on the back axle. On the contrary, the front wheels are steered and their rotation radiuses do not lie on the front axle [7].

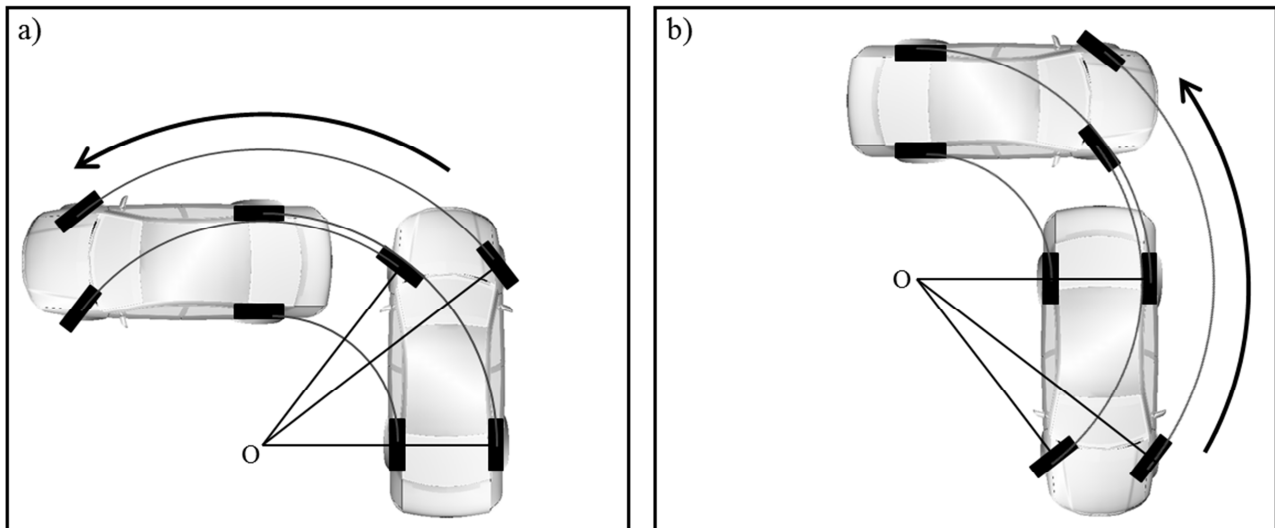


Fig. 1. Trajectory of a car turning forward (a) and backwards (b)

Moreover they are not collinear and hence the forward wheels are not parallel (the inner wheel is rotated of a higher angle than the outer one). The difference between features of the front and back axle is the reason why the front of the car shifts much more while turning than its back. The higher is the angle of wheels turn, the greater is the difference [8].

For comparison, turning of a car backwards has been presented on Fig. 1, b. The black curves on both sides of the figure indicate the direction of movement. Because of the reasons mentioned above, this movement differs from turning forward. Even though the car starts from the same position as before (the bottom right corner) and makes the same maneuver, it ends up in a different place. In the first situation the final position is the left side of the image, whereas in the second one it is the top-right corner of the image. This difference is the reason why parallel parking is performed by going backwards.

The car is equipped with three pairs of proximity sensors placed in its corners [9]. The sensors measure the distance from the closest object in one of three directions perpendicular to the car walls: front, right and back side of the car [10]. The black dots with numbers on Fig. 2 show the points of placement of the sensors and the arrows indicate the direction of distance measurement.

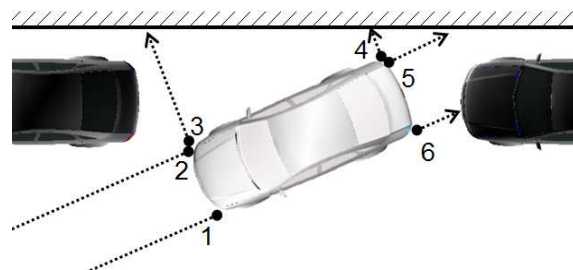


Fig. 2. Placement and direction of the car sensors

### 3. Parking Algorithm

The system can perform two kinds of parallel parking. In the first one, so called „one-shot parking”, after finding a spot the car moves only backwards (turning left or right, but using only the reverse gear). In the second one, called „multi-movement” parking, the car can change the direction of movement (backwards/forward) multiple times. The first phase of the algorithm is searching for a free parking spot. This phase has been presented on Fig. 3. It begins when the driver going forward sees a parallel parking in front of them and decides to park the car in it. They stop the car and turn on the parking system. Fig. 4 shows the flow chart of the parking system algorithm. In each iteration of the loop the car moves forward of a given, relatively small entity (e.g. one centimeter) and measures the length of the parking spot. In the measurement procedure the car takes the readout  $W$  of sensor 3 and compares it with the minimum spot width  $W_{min}$ . If it is lower

(Fig. 3, a), no parking space have been found ( $L=0$ ) and the algorithm performs another iteration of the loop. If it is greater, a space between parked cars has been found and measuring of its length starts.

The procedure repeats by adding the driven distance in the iteration to the currently found spot  $L$  (Fig. 3, b). When the found spot is longer than the minimum length for one-shot parking  $L_{min}$ , this maneuver is performed.

A possibility of multi-movement parking is checked only when measuring of the place has finished and one-shot spot has not been found. That means, that even if enough of space for multi-movement parking has been found, this maneuver is not performed immediately, but looking for a one-shot parking spot continues. One-shot parking has priority over multi-movement parking because it is much easier and quicker, even though it requires a little more of time to find a spot. The parking system uses three types of movement in every direction (forward or backwards): going straight, turning left and turning right. Turning of the car is performed with the maximum angle of rotation of the steering wheels.

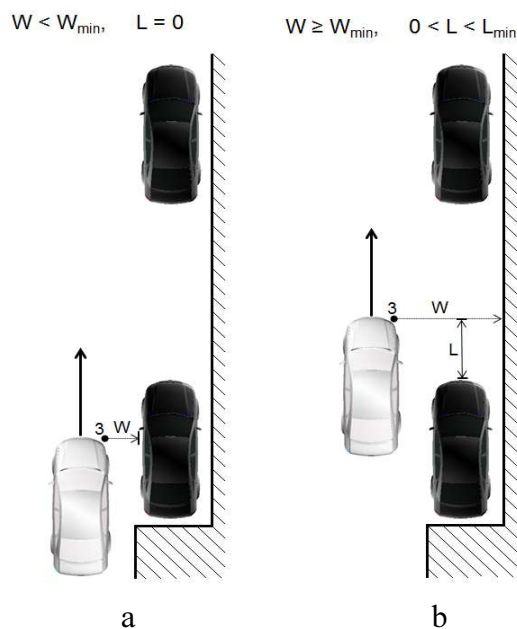


Fig. 3. Search for a parking spot before (a) and after (b) finding a space between cars

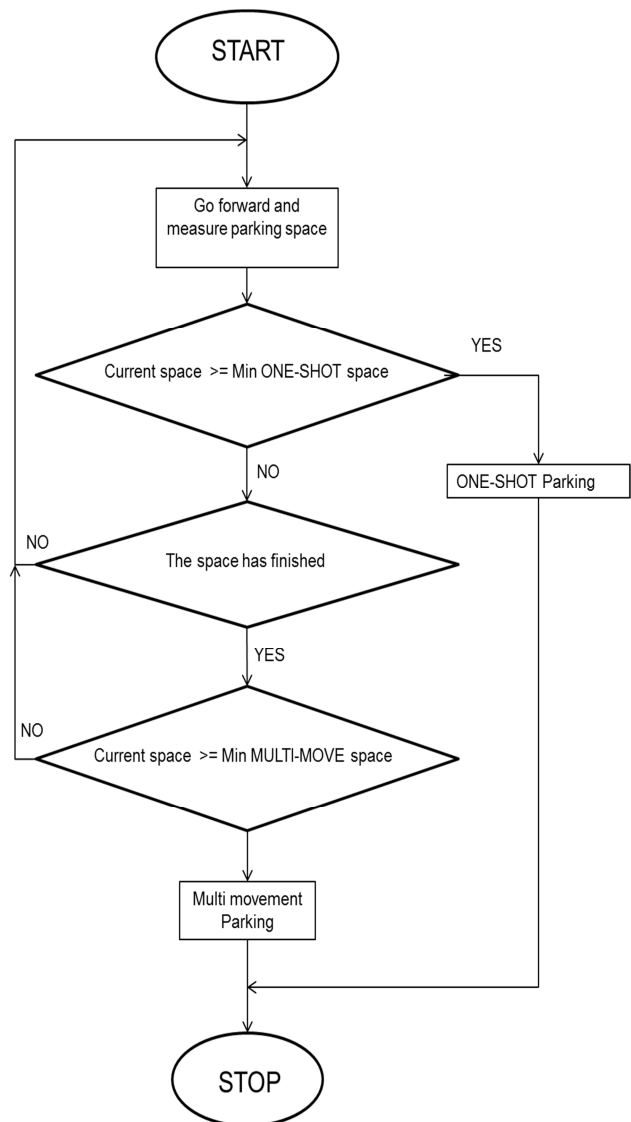


Fig. 4. Flow chart of the parking algorithm

#### 4. One-shot parking

One-shot parking is composed of three steps:

1. Turning right backwards
2. Going straight backwards
3. Turning left backwards

All three steps of parking are shown in Fig. 5. The numbers 1-4 denote possible positions of the car between adjacent steps. Note that position 2 appears in two phases: it is the end position of the first phase (Fig. 5, a) and the starting position of the second one (Fig. 5, b). Similarly, position 3 of the car is present in Fig. 5, b and Fig. 5, c.

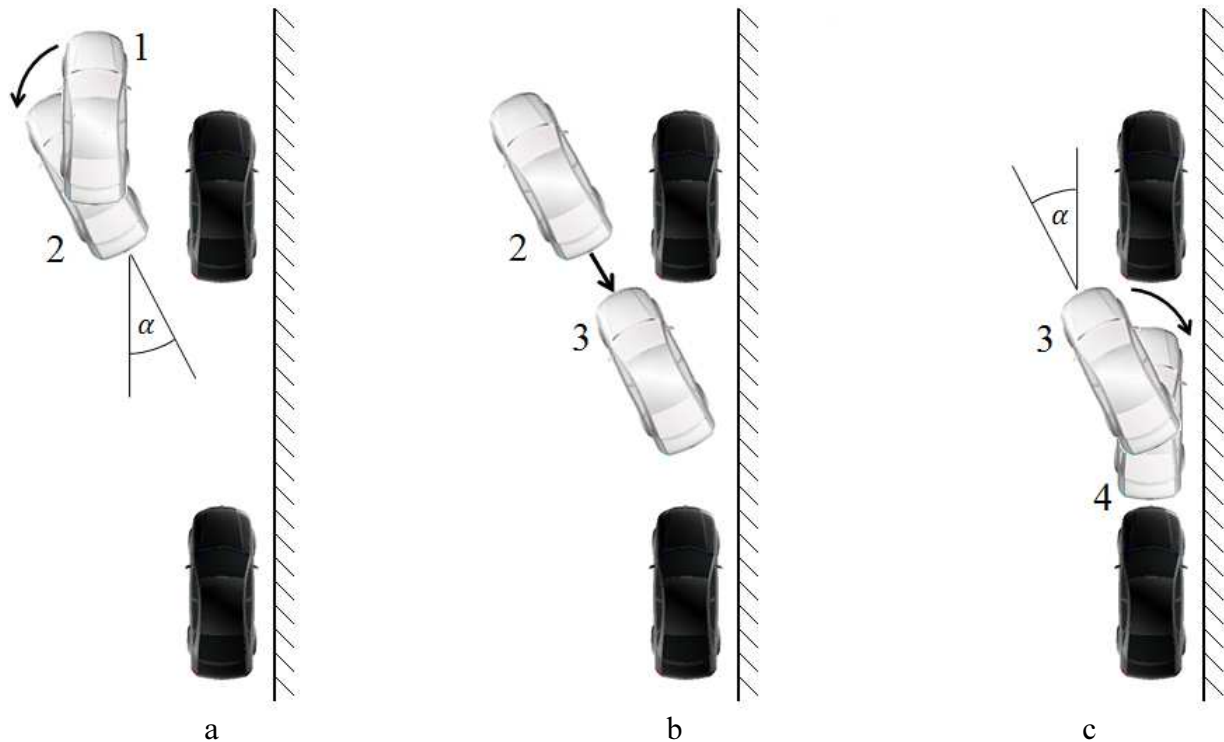


Fig. 5. The phases of one-shot parking:

- a) Turning left backwards; b) Going straight backwards; c) Turning right backwards

The minimum length of a one-shot parking spot is a place, in which a car can drive in performing only one turn backwards. In this case, the minimum length depends on the dimensions of the car and on the maximum angle of turn of the wheels  $\alpha$  (Fig. 6).

$$R = \frac{x}{\tan \alpha}$$

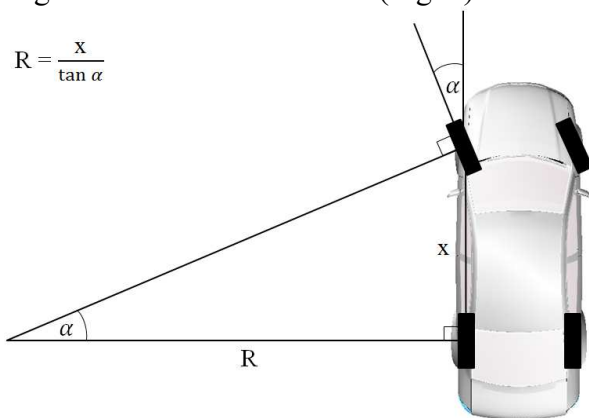


Fig. 6. Calculation of the radius R of rotation of the back left wheel

Turning starts from the end position of the previous maneuver – going straight backwards. This position is unknown in advance, however, it may be determined from the goal position of parking, which is known. In this position the car is parallel to the street and standing right behind the previous car (Fig. 5, c). The algorithm uses the fact, that car movements are reversible. That

is, if a car turns forward of a given angle and then turns backwards of the same angle, it ends up in the same place in which it started the maneuver. Hence, the end position of going out is the starting point of the third phase of parking.

Fig. 7, a shows determination of the one-shot parking length. A car going out turns left until its front shifts enough to be able to go straight forward without hitting the next car. This happens, when the front right corner of the car moves from point D to point A in the figure. The dotted line, parallel to the car after turning, demonstrates that the car is able to go straight forward.

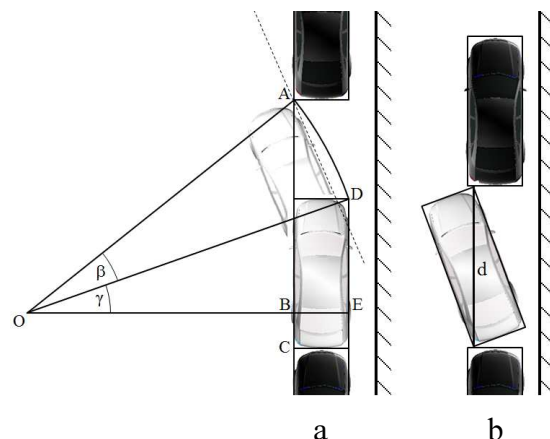


Fig. 7. Determination of minimum spot length for two kinds of parallel parking: a – one-shot; b) multi-movement

The required spot is the length of the  $|AC|$  segment, which is the sum of  $|AB|$  and  $|BC|$  segments.  $|BC|$  is the distance between the back axle and the back edge of the car and is known in advance.  $|AB|$  may be determined from the Pythagorean theorem, since the, angle  $\angle OBA$  is a right angle:

$$|OB|^2 + |BA|^2 = |OA|^2$$

$$|BA| = \sqrt{|OA|^2 - |OB|^2}.$$

$|OB|$  is the radius  $R$  of rotation of the back left wheel and is calculated using the trigonometric function tangent and the car data: the maximum turn angle of the inner wheel  $\alpha$  and the distance between the two axles  $x$  (Fig. 6).

The movement of the car in each step of parking depends on the movement in the next step. Therefore, the data is calculated in the opposite order, from the third step to the first one. As well as in case of determination of minimum spot length, reversibility of the car movement is used for calculations.

The angle  $\beta$  that the car rotates in the third phase is equal to the angle of rotation while going out of the parking. This angle is calculated based on (Fig. 7, a). Using the markings on the figure  $\beta$  is the difference between the angle  $(\beta + \gamma)$  and  $\gamma$ , obviously:

$$\beta = (\beta + \gamma) - \gamma$$

Both angles on the right side of the formula may be determined using inverse trigonometric functions:

$$(\gamma + \beta) = \arctan \frac{|AB|}{|OB|}$$

$$\gamma = \arctan \frac{|DE|}{|OE|}.$$

Segments  $|AB|$  and  $|OB|$  have been already determined. Segment  $|DE|$  is the distance from the back axle to the front edge of the car and is known from the dimensions of the car. Segment  $|OE|$  is the sum of the car width and segment  $|OB|$ . Finally:

$$\beta = \arctan \frac{|AB|}{|OB|} - \arctan \frac{|DE|}{|OE|}.$$

In the second phase of parking the car moves straight backwards from position 2 and stops at position 3. In these two points the direction of the car is the same. Position 2 is

determined from the first phase of parking that is turning left backwards. Since the parked car is parallel to the car before parking, the angles of rotation in the first and third phase are equal (Fig. 5, a) and (Fig. 5, c). To describe the position of the car it is enough to determine coordinates of one point belonging to the car. For convenience the center of the back axle has been chosen, since at position 1 it has the same y-coordinate as the center of rotation. Given the angle  $\beta$ , the segment  $f$  is determined (Fig. 8, a) as:

$$f = r \cos \beta.$$

Hence, given the radius of rotation, the x-coordinate  $x_2$  of the car in position 2 may be found. The shift  $s_x$  along the X-axis is calculated as the difference between x-coordinates of the car in positions 2 and 3. Then, the distance  $s$  driven by car in the second step of parking is determined, as well as the shift  $s_y$  along the Y-axis. The calculation is based on (Fig. 8, b) and trigonometric functions. When the coordinates  $(x_2, y_2)$  of the car in position 2 are found, the y-coordinate  $y_1$  of the car at position 1 is determined as (Fig. 8, a):

$$y_1 = y_2 + h,$$

where  $h$  is a side of the triangle on Fig. 8, a and is calculated using the tangent function.

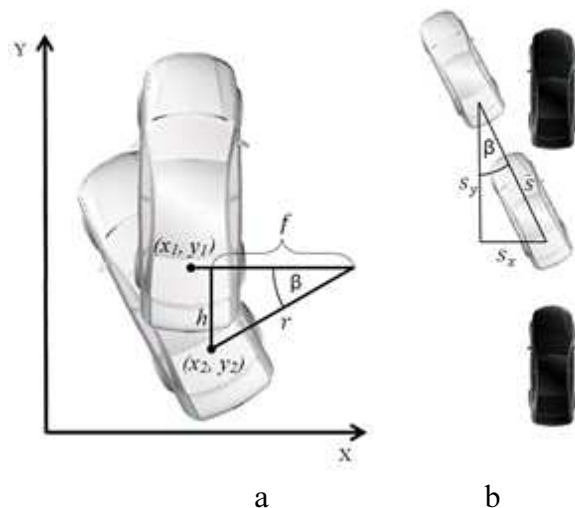


Fig. 8. Determination of the starting and end position in the first step of parking (a) and the distance driven in the second one (b)

### 5. Multi-movement parking

A car performing multi-movement parking can change the direction of movement several

times. The maneuver can be divided into three steps, as well as in case of one-shot parking. The two first phases are the same as before and the difference is only in the last one. Fig. 9 presents movement of a car in the third step of multi-movement parking. The arrows show the direction of rotation of the car and the letters on them indicate whether the car moves forward (F) or backwards (B). There is not enough of space for parking with only one turning left back, therefore the car is turning right forward. However, after that the car is not parallel to the street and needs another turning back.

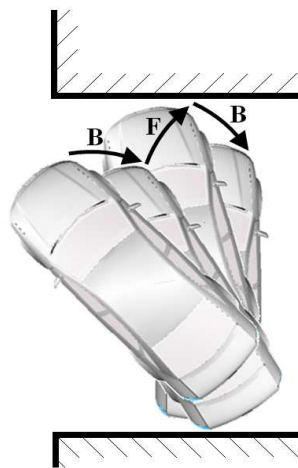


Fig. 9. Car parking with multiple change of move direction

Since the vehicle is allowed to change the direction of movement as many times as needed, the minimum length of the spot depends only on the car dimensions, and not on the maximum angle of turn of the wheels. The minimum length is the maximum segment included in the car projection and it is its diagonal (Fig. 7, b).

Calculation of the number of moves in the multi-movement parking is difficult, especially for a low angle of wheel turn and for a short parking spot. Without it, the starting position and direction of the car in the third step of parking cannot be determined. Therefore, a simulation of going out of a parking spot is performed. The method, as the previous one, is based on the reversibility of car movements. It simulates alternating of forward and backwards turns of a car until it is able to go straight forward without hitting any obstacles. The final position and rotation of the car after the

simulation is the starting position and rotation in the third step of multi-movements parking. Determination of the data in the previous two phases is the same as in case of one-shot parking.

### 6. Animation of the parking system

The developed parking system is presented as a 3-D animation [11]. The algorithm is programmed in C# language using DirectX API. The car in the animation makes movements as a real vehicle, that is moves either straight or rotates around a center calculated according to the formulas developed in the previous paragraphs. All four wheels rotate while the car moves and the two front wheels rotate along its Z-axis while steering.

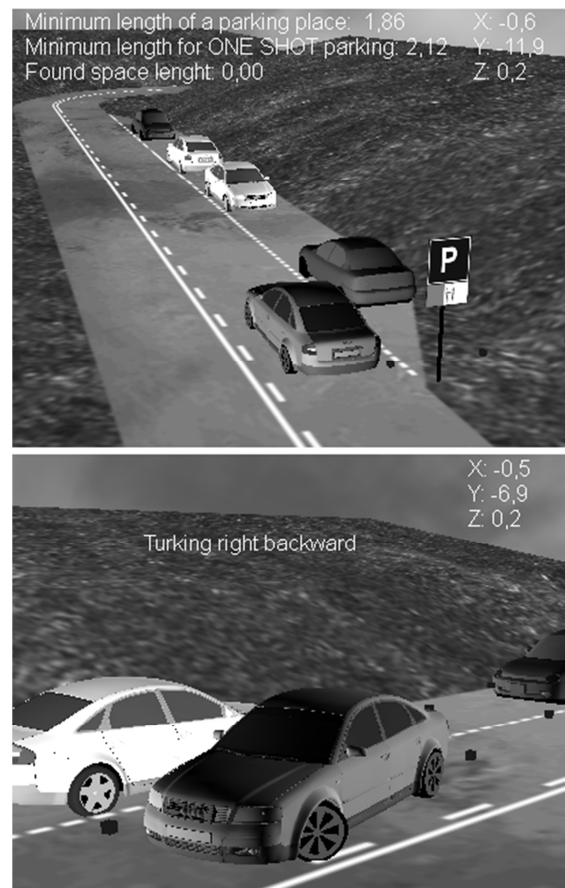


Fig. 10. Screenshots of the 3-D animation of the parking system

The animation shows both types of parallel parking. First, the car performs a one-shot parking, then goes out, moves forward to another spot and parks using the multi-movement method. The application lets the user steer the camera position and change the animation speed [12]. It is also possible to set

the maximum angle of wheels turn. That allows seeing how the size of the parking spot change and how the parking is performed e.g. for a very low angles of wheels turn. Fig. 10 shows two screenshots of the animation: searching for a parking space (on the top) and one-shot parking (on the bottom).

### 7. Conclusion

The developed parking system performs two kinds of parallel parking, depending on the available parking space. The algorithm is based on mathematical principles of movement of a vehicle. The developed 3-D animation is a pleasant evaluation of the system and shows that it works in practice. In the future the system could be applied to real vehicles and compared with the existing ones.

### References

1. Paromtchik I., and Laugier C., (1996), Motion Generation and Control for Parking an Autonomous Vehicle, *Proceedings of the IEEE International Conference on Robotics and Automation*. Minneapolis, MN, USA. pp. 3117 – 3122.
2. Paromtchik I., and Laugier C., (1996), Autonomous parallel parking of a Nonholonomic Vehicle, *Proceedings of the 1996 IEEE*, Tokyo, Japan. pp. 13 – 18.
3. Lo Y.K., Rad A.B., Wong C.W., and Ho M.L., (2003) Automatic Parallel Parking, *Intelligent Transportation Systems*, pp. 1190 – 1193.
4. Pohl J., Sethsson M., Degerman P., and Larsson J., (2006), A semi-automated parallel parking System for Passenger cars, *Transportation Science & Technology*, pp. 21 – 33.
5. Heravi O., Fogelberg M.J., and McMorris M.L., (2002), Control for Vehicle Differential, *US Patent 6,334,832*, pp. 8 – 11
6. Hatano M., (1988), Four wheel Drive vehicle with inter-axle Differential having dual Planetary gear sets, *US Patent 4,779,699*, pp. 4 – 8.
7. Jantos J., (1999), Drive Control in the Passenger car, *Journal of KONES*, Vol.6, No. 1-2, pp. 48 – 57.
8. Jain K.K., and Asthana R.B., (2002), *Automobile Engineering*, New Delhi, USA, pp. 305 – 306.
9. Multiple authors, (2016), Parking sensor, [Electronic resource], Available at: [https://en.wikipedia.org/wiki/Parking\\_sensor](https://en.wikipedia.org/wiki/Parking_sensor) (Accessed 30.05.2016).
10. Truesdell D., and Mariner E.T., (1993), Vehicle Proximity Sensor, *US Patent 5,229,975*, pp. 5 – 7.
11. Luna F, (2012), Introduction to 3D Game Programming with DirectX 11, pp. 37 – 53.
12. Woldgang E., (1996), Direct3D SHADERX: Vertex & Pixel Shader Tips and Techniques, pp. 30 – 45.

Received 04.06.2016



### Żygadło

Bartosz, MSc.  
Wroclaw University of  
Technology, Poland  
Minkowice Oławskie.  
55-220, Kościuszki 88.  
E-mail:  
bartosz.zygadlo@tlen.pl