

AUTONOMOUS PARKING SYSTEMS AND METHODS FOR VEHICLES**CROSS-REFERENCE TO RELATED APPLICATION**

[0001] This application claims the benefit of U.S. Provisional Application Ser. No. 62/949,451, filed on December 18, 2019, which is herein incorporated by reference in its entirety for all purposes.

FIELD OF THE INVENTION

[0002] The present disclosure relates to systems and methods for automatically parking a vehicle.

BACKGROUND

[0003] Autonomous driving (AD) has emerged to enable vehicles to navigate from one location to another with no human input. AD is achieved by including sensors to capture data which is analyzed by advanced data algorithms to evaluate the vehicle surroundings and making appropriate decisions for navigation.

[0004] Salient to AD is autonomous parking. To park a vehicle autonomously, it is required to understand the parking slot position, dimensions, and occupancy status. Conventional autonomous parking techniques employ sensors, such as Lidars, ultrasonic range detectors, radars, and/or cameras. Although such conventional techniques can find a suitable parking space or slot, they are unable to determine the correct positioning and boundary of a parking slot. The demarcation of a parking slot indicates the area allocated for a vehicle. Failure to accurately determine the boundary of a parking slot may lead to an improper parking maneuver, resulting in occupying more than one slot and or in a wrong orientation. Additionally, conventional techniques have problems handling variations in obstacles as well as variations in the environment.

[0005] From the foregoing discussion, it is desirable to provide an autonomous parking system and methods which can accurately determine a parking slot despite variations in obstacles and the environment with real-time performance.

SUMMARY

[0006] Embodiments generally relate to systems and methods for autonomous parking systems for vehicles. In one embodiment, the system includes an autonomous parking system for a vehicle including a detection subsystem having an image capture module for capturing images to provide a camera stream, and a processor module. The processor module receives the camera stream and processes the camera stream. The processor module is configured to process frames of images including processing a current frame and a previous frame of the camera stream to identify a parking slot, to check the parking slot to determine if it is occupied using motion analysis, and if unoccupied, to validate the unoccupied parking slot to determine if the unoccupied parking slot is suitable for the vehicle.

[0007] In another embodiment, a method for autonomously parking a vehicle includes performing motion analysis of a current image frame and a previous image frame from an image capture module to determine a parking slot, checking the parking slot to determine if the parking slot is occupied, and validating the parking slot if the parking slot is unoccupied based on a vehicle size and a parking slot size.

[0008] These and other advantages and features of the embodiments herein disclosed, will become apparent through reference to the following description and the accompanying drawings. Furthermore, it is to be understood that the features of the various embodiments described herein are not mutually exclusive and can exist in various combinations and permutations.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] In the drawings, like reference characters generally refer to the same parts throughout the different views. Also, the drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of various embodiments. In the following description, various embodiments of the present disclosure are described with reference to the following, in which:

- [0010] Fig. 1a shows a simplified block diagram of an autonomous parking system;
- [0011] Fig. 1b shows an image capture module of the detection subsystem;
- [0012] Figs. 2a-b show a general process flow for autonomous parking;
- [0013] Fig. 3 shows an embodiment of a calibration process for camera units;
- [0014] Fig. 4 shows a simplified general process flow for identifying a parking slot;
- [0015] Fig. 5a shows a process flow for identifying and determining the status of a parking slot;
- [0016] Figs. 5b-c show examples of a motion analysis;
- [0017] Fig. 6a shows a process flow for determining ground motion estimation based on apparent motion;
- [0018] Fig. 6b depicts an example of perspective rectified motion analysis;
- [0019] Fig. 6c shows an exemplary histogram from calculated perspective rectified motion;
- [0020] Figs. 6d-e depict examples of ground motion estimation;
- [0021] Fig. 7a shows an embodiment of an occupancy check flow using HGMS;
- [0022] Fig. 7b depicts another embodiment of occupancy check flow using HGMS;
- [0023] Fig. 8 shows exemplary parking slot markings;
- [0024] Fig. 9a shows an embodiment of a pair of 1-D image filters;
- [0025] Fig. 9b shows a process flow of detecting parking slot marking using 1-D image filters;

- [0026] Fig. 10a depicts a process flow of parking slot marking detection;
- [0027] Fig. 10b depicts an embodiment of parking slot marking detection;
- [0028] Fig. 11a shows a comparison of occupancy check of the different techniques based on different input images; and
- [0029] Fig. 11b shows a table of scores for performance indicators for the different types of systems related to detecting a free parking slot.

DETAILED DESCRIPTION

[0030] Embodiments described herein generally relate to autonomous driving systems for a vehicle, such as a car. In particular, the embodiments relate to an autonomous parking system. The autonomous parking system includes parking slot detection and occupancy check using homography guided motion segmentation (HGMS).

[0031] Fig. 1a shows a simplified embodiment of an autonomous parking system 100 for vehicles, such as cars. Other types of motor vehicles may also be useful. As shown, the parking system includes a controller subsystem 120 and a detection subsystem 150. The detection subsystem includes a processor module 152 and an image capture module 154. As for the controller subsystem, it includes a vehicle information module 124, a path planner module 126, and a vehicle controller module 128. The subsystems are configured to detect a parking slot using, for example, the structure of parking lines, check on the occupancy status, and direct the vehicle into an unoccupied parking slot.

[0032] Regarding the detection subsystem, the image capture module captures images of potential parking slots. For example, the image capture module includes camera units for capturing images of potential parking slots. In one embodiment, the image capture module includes 2 camera units, one mounted on each side of the vehicle to capture images from the sides of the vehicle as it is traveling. The camera units stream images for processing by the processor module. Each camera unit provides a camera stream for processing. In one

embodiment, a camera stream includes images in which 2 consecutive images or image frames of the camera stream should capture the same position or object with small apparent motion while the vehicle is traveling. The capture rate, for example, depends on the speed of the vehicle. When a vehicle is searching for a parking slot, the speed of the vehicle may be about less than or equal to 10 km/hour. Other speeds may also be useful. The capture rate of the camera units may be, for example, about 20 frames per second for a vehicle travelling at about 10km/hour. Other capture rates may also be useful, depending on the speed of the vehicle. The camera units may be configured to capture images at varying rates based on vehicle speed.

[0033] A capture control unit may control the capture rate of the camera units based on the vehicle speed. The capture control unit may be part of the image capture module. Other configurations of the capture control unit may also be useful. For example, the capture control unit may be part of the processor module.

[0034] The processor module processes the captured images to identify an unoccupied parking slot. For example, the processor module may include a central processing unit (CPU) and memory for processing the images. In some embodiment, the CPU may be an 8-core CPU, such as the i7-7700 HQ CPU from Intel. Other types of CPUs may also be useful. In one embodiment, the processor module identifies an unoccupied parking slot using HGMS. In one embodiment, the processor module determines a parking slot using the structure of parking lines and the occupancy status check. The use of HGMS to determine an unoccupied parking slot can be achieved with the desired time and accuracy performance without the need for high-performance processors, such as a graphics processor unit (GPU). This reduces the costs of the autonomous parking system.

[0035] As for the controller subsystem, it provides information for the detection subsystem to determine an unoccupied parking slot based on vehicle information or data

received from the vehicle's system and to guide the vehicle into the unoccupied parking slot once it is identified. For example, vehicle information may include vehicle location and vehicle dimensions. Once a free slot is identified, the detection subsystem provides the slot location with respect to the vehicle location.

[0036] In one embodiment, vehicle data is provided by the vehicle's system or systems. For example, vehicle data may include ego motion values such as speed, direction of travel, and inertial measurement unit (IMU) values. In addition, current location and the size of the vehicle are provided. The vehicle data is provided to the vehicle information module. Other types of vehicle data may also be provided. The various information may be received from the vehicle's control system, global positioning system (GPS), Ego motion system as well as other systems.

[0037] The vehicle information module passes vehicle data to various modules of the controller subsystem as well as to the detection subsystem. For example, vehicle data is passed to the processor module of the detection subsystem as well as the path planner module and the vehicle controller module of the controller subsystem.

[0038] Once a parking slot is detected, the detection subsystem uses the vehicle data, such as the size of the vehicle, to determine whether the parking slot is suitable for the vehicle. For example, the detection subsystem determines whether the vehicle can fit into the dimensions of the parking slot. If the slot is suitable, it is validated as an available parking slot. In addition, vehicle speed may be provided to the detection subsystem to control the capture rate of the camera units.

[0039] Once a parking slot is validated, the information related to the parking slot is provided to the path planner module. For example, the GPS location of the validated parking slot, width and length of the parking slot, as well as the distance and orientation from the current vehicle location are provided to the path planner. Based on the information received,

the path planner generates a path for the vehicle to the validated parking slot, including angle (perpendicular) or parallel parking of the vehicle based on the slot orientation.

[0040] The path to the validated parking slot is passed to the vehicle controller module. Based on the path, the vehicle controller module sends control signals to the vehicle to guide the vehicle into the validated parking slot. The information may include speed, steering angle and braking information.

[0041] Fig. 1b shows an image capture module 154 of the detection subsystem. As shown, the image capture module includes camera units 158 mounted on the sides of the vehicle. A camera unit includes a camera. Preferably, the camera of a camera unit is a wide-angle camera. For example, the wide-angle camera may have a 120° wide-angle lens. Other angles for the wide-angle lens may also be useful. In one embodiment, a camera unit is mounted with a pitch and at a position on the vehicle which enables capturing of an area between about 0.3 to 5 m. For example, a wide-angle camera is mounted at a position at or near the side view mirrors of the vehicle with a pitch of about 20°. Other positions and pitch configurations for capturing other sized areas may also be useful.

[0042] The resolution of the camera should be adequate to capture and process the images to determine an unoccupied parking slot with sufficient accuracy within the desired time performance requirements. The camera resolution may be programmable. In one embodiment, the resolution of the camera may be about 150k to 250k pixels. For example, the resolution may be about 416 x 416 or 640 x 360. Other camera resolutions may also be useful. The desired resolutions may depend on processor speed and time performance requirements. The higher the resolution, the longer it takes to process for a given processor performance.

[0043] Figs. 2a-b show a simplified process flow 200 for auto parking a vehicle using the autonomous parking system. The autonomous parking system can be operated when the

vehicle is in autonomous driving mode or not. For example, the vehicle may be autonomously driven without driver aid or driven by the driver manually. The process starts at 210. For example, auto parking mode is triggered. In auto parking mode, the vehicle, for example, may be in the parking area.

[0044] At 215, the autonomous parking system is initialized. For example, the various subsystems and modules are initialized for auto parking operation, such as the camera units and software components. At 220, the auto parking search mode is initiated. This causes the car to be driven in autonomous mode in search of a parking slot. For example, the vehicle controller module may cause the vehicle to drive at a slow speed, such as at about 10 km/hr. In addition, the path planner module may plan a route for the car to proceed in search of a parking slot. For example, the path planner routes the vehicle to available slots based on current map information.

[0045] At 225, the image capture module captures images from the sides of the vehicle as it is traveling and sends the captured images as a camera stream to the processor module for processing. The processor module processes the camera stream from the camera units at 230. For example, the processor module processes two successive image frames of the camera stream using HGMS to detect an unoccupied parking slot. For example, apparent motion (AM) analysis is performed on two successive or sequential image frames to detect an unoccupied parking slot. This includes, in one embodiment, motion analysis to detect a parking slot and its occupancy status. In the event the parking slot is unoccupied, details of the unoccupied parking slot are determined. Details of the parking slot may include the dimensions and position of the parking slot with respect to the current vehicle position as well as parking slot orientation.

[0046] The processor module validates the unoccupied parking slot at 235. For example, the processor module determines if the vehicle fits into the unoccupied parking slot

based on vehicle dimensions received from the vehicle information module. In addition, the processor module determines how to best park the vehicle, such as perpendicularly, diagonally, or parallelly. For example, information is provided for use in path planning by the path planner module.

[0047] If the parking slot is not validated at 240, the process returns to 225. For example, the process returns to 225 if the parking slot is not suitable for the vehicle, such as the dimensions of the vehicle is larger than the dimensions of the parking slot. On the other hand, if the parking slot is validated, the process continues to 245.

[0048] At 245, the process determines if the vehicle is in autonomous vehicle (AV) mode or not. For example, the path planner module determines if the vehicle is in AV mode or not. If the vehicle is in AV mode, the process continues to 255. If the vehicle is not in AV mode, the system queries the user for confirmation at 250 to park in the validated parking slot. For example, the path planner module queries the user. If the user confirms parking in the validated parking slot, the process continues to 255. If not, the process returns to 225.

[0049] At 255, the path to park the car in the validated parking slot is computed. For example, the path planner module computes the path for parking the vehicle into the validated parking slot based on information of the validated parking slot. In one embodiment, the optimal trajectory for auto parking, from the vehicle location to the parking slot and orientation of the parking slot, is computed.

[0050] The vehicle, at 260, is navigated into the parking slot. The vehicle controller module, for example, provides the control signals to perform auto parking at the desired speed (longitudinal) control and steering (lateral) control based on the computed path. After parking is completed, the driver is notified by the autonomous parking system of successfully parking the vehicle at 265. The autonomous parking system shuts down at 270, for example, releasing the auto parking resources. The auto parking mode terminates at 275.

[0051] HGMS involves motion analysis of successive images when the vehicle is moving. For example, apparent motion is computed using dense or semi-dense optical flow between two successive image frames of a camera stream. For example, apparent motion analyzes the movement or flow of corresponding pixels between two successive image frames. Due to the perspectivity of the camera, the same amount of motion in the real world results in a different magnitude of pixel motion at different rows of the image. This is based on the motion parallax principle.

[0052] In one embodiment, calibration is performed to determine a perspectivity rectified factor for each row to determine the perspectivity relationship of each row of the image to the real world and saved as perspectivity ratio map (PR). For example, $PR(y)$ indicates the perspectivity relation at a given row (y^{th} row) on an image with h rows, where y is from 0 to $h - 1$. The perspectivity ratio map indicates the scaling effect of perspectivity on a unit length in the world system at each row of the image. $PR(y)$ helps to rectify the perspectivity across different rows of the image.

[0053] The calibration generates a homography matrix $h.Mat$, which may be a 3 by 3 matrix containing homography transformation between camera and real world. For example, $h.Mat$ represents the transformation between two planes using rotation and translation relationship. The homography matrix $h.Mat$ facilitates image coordinate to world coordinate mapping and vice-versa. To improve performance, lookup tables are generated for image-to-world coordinate mapping and world-to-image coordinate mapping.

[0054] Fig. 3 shows an embodiment of a calibration process 300. As shown, a calibration pattern 310 of a known dimension is placed on the ground at a known distance from a camera unit. Ground points with known distances are mapped to their corresponding image points. This helps to generate a homographic transformation. Exemplary range of ground or reference points are within 5 m longitudinally and 2 m laterally. For example, the

calibration pattern is a rectangular-shaped pattern. The known width of the pattern is Dwld. Non-rectangular shaped patterns may also be useful.

[0055] An image 320 of the calibration pattern 310 is generated from the camera unit. The image, for example, includes $h \times t$ pixels. For example, the image has h rows of pixels and t columns of pixels. The top of the image may be the far side of the image with respect to the vehicle and the bottom of the image may be the near side of the image with respect to the vehicle. The top of the image may correspond to the first row of pixels ($y = 0$) and the bottom of the image may correspond to the last row of pixels ($y = h - 1$). Other configurations may also be useful.

[0056] The sides of the calibration pattern in the image are extended from the top to the bottom of the image. For example, left and right side lines 330l and 330r extend from the sides of the calibration pattern to the top and bottom of the image. As can be seen, Dwld on the far side is narrower than Dwld on the near side due to the perspectivity of the camera. As such, the width or distance between the two side lines at the top or far side of the image, which is referred to as Fpix is narrower or smaller than the width or distance between the two side lines at the bottom or near side of the image, which is referred to as Npix. The Perspectivity Ratio Map (PR) for each row is calculated based on the row scale factor (RowScaleFact) as defined below:

$$\text{RowScaleFact} = (\text{Npix} - \text{Fpix})/h-1 \quad \text{Equation (1).}$$

The PR for the y^{th} row between rows 0 to $h-1$ is calculated as follows:

$$\text{PR}(y) = \text{Npix}/(\text{Fpix} + y * \text{RowScaleFact}) \quad \text{Equation (2).}$$

The perspectivity rectified factor for each row is calculated and forms the Perspectivity Ratio Map (PR).

[0057] The calibration and generation of the homography matrix $h.Mat$ are performed for both camera units. For example, the right and left camera units of the image capture module are calibrated and respective homography matrices are generated.

[0058] Fig. 4 shows a simplified general process flow 400 of an embodiment for identifying a parking slot. As discussed, identifying a parking slot involves motion analysis on 2 successive image frames of a camera stream. The image frames, prior to the analysis, are converted to grayscale at 410. For example, color images are converted to grayscale prior to motion analysis for detecting a parking slot at 420. The motion analysis, in one embodiment, is performed to determine apparent motion based on dense optical flow between two successive image frames of a camera stream. For example, apparent motion analyzes the movement or flow of corresponding pixels between two successive image frames. The parking slot detection is based on parking slot lines.

[0059] As discussed, the camera units are calibrated at 430 to set up the system and homography information is generated at 440 from the calibration. The homography info is provided to provide image-to-world mapping for the motion analysis at 450. Once a parking slot is identified, the process performs the occupancy status check at 460.

[0060] Fig. 5a shows a simplified process flow 500 for identifying a parking slot or space. Identifying a parking slot involves analyzing 2 consecutive image frames captured by the image capture module. The images, if in color, are converted to grayscale. Apparent motion analysis is performed on the 2 consecutive image frames at 510. Apparent motion analyzes the movement of corresponding pixels between the 2 consecutive or successive image frames taken from the vehicle when moving during auto parking mode. Apparent motion is computed using dense optical flow between two successive image frames.

[0061] At 520, ground motion estimation is computed based on the results of the apparent motion. For example, ground motion is used to segment the apparent motion into

ground and obstacles. After ground motion estimation is computed, the process checks to determine if obstacles are present or not at 530. If no obstacles are identified, then a free parking slot is found and process proceeds to 540. If an obstacle is identified, then the parking slot is not available or free and analysis continues. For example, the current image frame is compared to the next image frame to analyze. For example, the current image frame becomes the previous image frame and the next image frame becomes the current image frame.

[0062] At 540, the boundary of the free parking slot is refined. For example, local ground motion correction is performed to refine ground motion. This helps in handling ground surface inclinations and minor camera view changes to extract accurate object boundaries. Parking slot marking detection is performed at 550. The occupancy status and details of the free parking slot are provided, for example, for validation and subsequent processes.

[0063] Fig. 5b shows an example of a motion flow analysis 511. As shown, 2 successive image frames 515 and 525 of an occupied parking slot in grayscale are provided for motion analysis. The motion analysis employs dense optical flow to determine the apparent motion flow 535 of each pixel of the two consecutive image frames.

[0064] Fig. 5c shows another example of a motion flow analysis 512. As shown, 2 successive image frames 516 and 526 of an unoccupied parking slot in grayscale are provided for motion analysis. The motion analysis employs dense optical flow to determine the apparent motion flow 536 of each pixel of the two consecutive image frames.

[0065] Fig. 6a shows a process flow 600 for determining ground motion estimation based on apparent motion. As discussed, apparent motion is the movement of corresponding pixels between two image frames of a moving camera. Apparent motion is computed using dense optical flow between two successive image frames. Apparent motion is used to

determine perspectivity rectified motion at 610. Perspectivity rectified motion (PRM) can be computed by applying the perspectivity correction or perspectivity rectified factor on the apparent pixel motion (AM). The perspectivity rectified factor is determined based on camera calibration and stored in the homography matrix *h.Mat*. The PRM of a specific pixel at the location *x, y*, or PRM(*x,y*) can be calculated as follows:

$$\text{PRM}(x,y) = \text{AM}(x,y) * \text{PR}(y) \quad \text{Equation (3),}$$

where

AM(*x,y*) is the apparent motion of a pixel at location *x, y*, and PR(*y*) is the perspectivity rectified factor at row *y*. AM is calculated for all pixels of the image.

[0066] At 620, a motion histogram is computed. The motion histogram is a plot of PRMs of all pixels of the image. The range of the histogram includes positive (+ve) and negative (-ve) motions. The histogram is smoothened to handle local minima. The peak of the histogram and its corresponding minima, for example, before and after the peak, is determined at 630. This indicates the PRM range exhibited by a majority of the pixels. This corresponds to ground motion.

[0067] Fig. 6b depicts an example of perspectivity rectified motion analysis 606. As shown, previous and current image frames 616 and 626 are analyzed. For example, the grayscale images of the previous and current image frames are analyzed. The analysis includes motion analysis to determine apparent motion 636 of corresponding pixels from the previous and current image frames. Based on apparent motion, perspectivity rectified motion flow 646 of the pixels are calculated by the perspectivity rectified factor to the apparent pixel motion (AM) for each corresponding pixel pair of the previous and current image frames.

[0068] Fig. 6c shows an example of a motion histogram 615. The motion histogram is calculated from perspectivity rectified motion, as described in Fig. 6. For example, the histogram indicates the magnitude of motion in the image at different pixels. A majority of

ground zone is seen in the image along with any potential obstacles. The histogram peak represents the perspectivity rectified motion magnitude exhibited by the majority of pixels, typically corresponding to ground.

[0069] Figs. 6d-e depict examples of ground motion estimation. Referring to Fig. 6d, the ground motion estimation 607 is of previous and current image frames 617 and 627 of an occupied parking slot. The analysis includes motion analysis to determine apparent motion 637 of corresponding pixels from the previous and current image frames. Based on apparent motion, perspectivity rectified motion flow of the pixels are calculated by the perspectivity rectified factor to the apparent pixel motion (AM) for each corresponding pixel pair of the previous and current image frames. A motion histogram 647 is calculated from perspectivity rectified motion flow.

[0070] As for Fig. 6e, the ground motion estimation 608 is of previous and current image frames 618 and 628 of an unoccupied parking slot. The analysis includes motion analysis to determine apparent motion 638 of corresponding pixels from the previous and current image frames. Based on apparent motion, perspectivity rectified motion flow of the pixels are calculated by the perspectivity rectified factor to the apparent pixel motion (AM) for each corresponding pixel pair of the previous and current image frames. A motion histogram 648 is calculated from perspectivity rectified motion flow.

[0071] Fig. 7a shows an embodiment of a process flow 700 for occupancy check using HGMS. For example, the occupancy check determines if the parking slot has an obstacle present or not. To extract obstacles, reference motion estimation is determined at 710. In one embodiment, homography guided reference motion estimation is employed.

[0072] A reference ground plane motion (RefGndMap) is calculated. The RefGndMap can be estimated by various techniques. For example, RefGndMap can be estimated from the peak motion computed in the ground motion estimation process (e.g., seed row is h-1), the

vehicle speed and calculating the equivalent pixel motion in the seed row ($h-1$) of the image, the motion computed at a seed row of the parking slot marking (a seed row is selected based on the parking slot position in the image), or a combination thereof. In one embodiment, ground motion estimation is calculated from the peak motion computed in the ground motion estimation process (e.g., seed row is $h-1$) and validated by the vehicle speed and calculating the equivalent pixel motion in the seed row ($h-1$) of the image and calculating the motion at a seed row of the parking slot marking (a seed row is selected based on the parking slot position in the image). From the reference motion calculated at one of the image rows, the RefGndMap for all image rows is calculated using PR. The reference ground plane is used to, for example, refine obstacle boundary. The reference ground plane motion represents the motion flow of the ground plane for the current pair of image frames (e.g., the previous image frame and the current image frame).

[0073] In one embodiment, RefGndMap is computed based on current vehicle speed and camera homography, such as $h.Mat$. For example, the displacement between the current and previous image frames in the world coordinate system is computed. Then, the equivalent pixel displacement at each image row is computed based on the homography matrix $h.Mat$ and the reference perspectivity ratio map PR calculated as part of the calibration process.

[0074] In some cases, the speed of the vehicle may not be available. In such cases, dense motion may be used. This includes selecting a seed row. A seed row is a row indicating the ground plane. For example, a slot marking pixel value may be used as a reference to select a seed row. The mean motion at the seed row (RefFlowSeedRow) is computed according to equation 4a below:

$$RefFlowSeedRow = 1/(ImgWidth) * \sum_{i=0}^{ImgWidth-1} flow(i) \quad \text{Equation (4a).}$$

Based on RefFlowSeedRow and PR, the motion at each image row j is computed according to equation 4b below:

$$RefGndMap_j = RefFlow_{GroundRow} * PR_j / PR_{GroundRow} \quad \text{Equation (4b).}$$

Equation (4b) can be applied to any of the techniques used to determine RefGndMap.

[0075] In another embodiment, since ground motion is estimated globally, RefGndMap can be computed based on pixels corresponding to the ground from the global histogram. For each row, pixels corresponding to the ground are extracted and a row motion histogram (row_motion_histogram) is created. For example, a row_motion_histogram is created for each row. Peak motion of the row_motion_histogram is computed and used as the ground reference motion for the corresponding row.

[0076] At 720, flow computation in the parking slot region is calculated based on the current and previous image frames. For example, flow computation is achieved using dense optical flow to generate the dense flow map. For example, apparent motion flow is calculated. The apparent motion flow may be calculated previously with respect to determining perspective rectified motion.

[0077] At 730, reference motion compensation is performed. Reference motion, in one embodiment, is compensated by subtracting RefGndMap from the dense flow map. Reference motion compensation results in the obstacle motion (ObstFlow) as shown in equation 5a below:

$$ObstFlow_{i,j} = flow_{i,j} - RefGndMap_j \quad \text{Equation (5a),}$$

where

ObstFlow_{i,j} is the obstacle flow at pixel location i, j,

flow_{i,j} is the dense flow of the pixel at location i, j, and

RefGndMap_j is the RefGndMap at row j.

[0078] To handle any noise in the flow computation, homography guided adaptive threshold (AdptThdMap_j) is applied at 740. AdptThdMap_j is defined according to equation 5b below:

$$AdptThdMap_j = Threshold * PR_j$$

Equation (5b),

where

$AdptThdMap_j$ is the $AdptThdMap$ at row j ,

PR_j is the perspective rectified factor at row j , and

Threshold is the defined threshold

As such, $AdptThdMap_j$ is defined with perspective in view and gradually increases with row number.

[0079] $AdptThdMap_j$ produces binary masks of potential obstacle regions, as shown by equation 5c below:

$$ObstacleMap_{i,j} = \begin{cases} 1, & \text{if } ObstFlow_{i,j} \geq AdptThdMap_j \\ 0, & \text{otherwise} \end{cases}$$

Equation (5c).

For example, when $ObstacleMap_{i,j}$ is equal to 1, it corresponds to a non-ground pixel and when $ObstacleMap_{i,j}$ is equal to 0, it corresponds to a ground pixel. After grouping and analyzing the neighboring homogenous pixels, masks are created. White pixel mask represents object zone and black represents ground zone.

[0080] The binary masks are further post-processed at 750 with binary morphology closing operation to eliminate breaks due to smooth texture of any obstacle. A filter may be applied based on the pixel area to suppress noise blobs of small sizes and retaining the binary masks of the obstacles in the selected parking region in the current frame. For example, binary masks with value 1 represent the pixels which have motion higher than the reference ground motion. Such pixels represent the obstacles. Based on the size of the mask, we can estimate the size of an obstacle can be estimated.

[0081] After postprocessing and size filtering, the process may proceed to determine occupancy status.

[0082] Fig. 7b depicts a sequence of occupancy check 701 based on HGMS. As shown, a current image frame 711 and a previous image frame 721 are captured by the image

capture module and processed. For example, the image frames are grayscale images. Dense motion analysis is performed to produce a motion flow 731. For example, dense optical flow is performed to produce the motion flow. Homography guided reference motion analysis is performed to produce the RefGndMap 741. Reference compensation is performed on the dense optical flow to produce obstacle flow 751. Adaptive thresholding is performed to produce obstacle masks in the masked obstacle flow 761.

[0083] In one embodiment, the autonomous parking system detects parking slots or spaces based on parking slot markings. Detecting parking slots based on other features may also be useful. For example, if no parking slot markings are present, the ground zone from the occupancy check can be analyzed for free space dimension and generate the parking space suitability. Parking slots are generally represented by a set of lines or parking slot markings. The color of the parking slot markings may be red, yellow, or white. Other colors may also be useful. The parking slot markings are used to indicate the length, width, and orientation of the parking slot allocated for each vehicle.

[0084] Fig. 8 shows exemplary types of parking slot markings. A parking slot 810 may include L-shaped markings located at corners of the parking slot. The L-shaped markings are configured to define the parking slot. Alternatively, a parking slot 820 may be defined by a set of lines configured as a rectangle defining the parking slot. In other cases, the parking slot may be defined by parallel lines along the length of the parking slot, such as shown by parking slots 830 and 840. The first ends of the parallel lines may include perpendicular lines. As shown, the perpendicular lines may be configured to form T-shaped first ends. The perpendicular line may alternatively be configured to form L-shaped first ends. Other configurations of first ends may also be useful. Parallel lines, for example, are used to define a row of adjacent parking slots. The parallel lines may be perpendicular to the row direction, as shown by parking slot 830. Alternatively, as shown by parking slot 840, the parallel lines

are disposed at an angle to the row direction. What is shown in Fig. 8 are only some examples of commonly employed parking slot markings. Other configurations of parking slot markings may also be applicable.

[0085] Since parking slot markings have similar structural characteristics as road lane markings, a gradient is employed to locate the parking slot markings and to validate the structural characteristics of the parking slot markings.

[0086] In one embodiment, a pair of 1-D image filters 900 are used, as shown in Fig. 9a. The filters 911 and 922 are employed to compute the rising and trailing edges of a grayscale image. For example, based on equation 6 below, each of the filters f is applied to the input gray image I to determine the candidate rising and trailing edge pixels:

$$f \circ I(x) = \sum_{i=-N}^N f(i)I(x+i) \quad \text{Equation (6),}$$

where

N is the length of the 1-D filter with exemplary range of about 1 to 3, and

x is equal to each pixel on the input image.

[0087] As an example, a valid parking slot marking may be a line mark having a combination of the rising and trailing edge pixels separated by a distance similar to the line mark thickness. The rising and trailing edge pixels are validated based on the condition that the rising pixel is followed by a falling edge pixel with a distance less than a threshold value. The center pixel of the line mark is computed by averaging each rising and trailing pixel position. This produces a binary image, where the bright pixels are processed to compute the approximate line segments, which are used for further processing.

[0088] As shown, the filters are configured for detecting vertical lines. Configuring the filters to detect horizontal lines may also be useful. Since the filters are configured to handle lines with angles from -45° to 45° , V-H pairs are not constrained to 90° .

[0089] Fig. 9b depicts a process 901 using the 1-D filters. A grayscale image 910 is provided for filtering. Image 920 depicts the effect of vertical lines after the filtering process using the 1D filters. For example, the edges of the vertical line marks are shown. Image 930 depicts the effect of the horizontal lines after the filtering process using the 1D filters. For example, the edges of the horizontal line marks are shown. As for image 940, it depicts the effect of the combination of vertical and horizontal line marks. For example, combined horizontal and vertical edges of the line marks are shown.

[0090] Fig. 10a shows a process flow 1000 for detecting parking slot markings or line marks defining the parking slot. The process is employed to determine the correct pair of slot corners for the parking slot, which is then used to determine the dimensions and orientation of the unoccupied parking slot for subsequent validation. The process analyzes an image. For example, the current image frame of the camera stream is analyzed. At 1010, the image frame is converted to grayscale. In the case the image is black and white or has already been converted, 1010 can be skipped.

[0091] The parking slot detection is configured to detect horizontal and vertical line marks. In one embodiment, the 1D filters are configured for detecting vertical line marks. Configuring the 1D filters for horizontal line marks may also be useful.

[0092] To detect horizontal parking slot markings or line marks of the image, the image is rotated at 1020 by 90°. For example, the horizontal line marks become vertical line marks. The process analyzes the image for vertical line marks or segments at 1030. For example, the 1D filters are used to identify vertical line segments. The center pixels of the vertical line segments are calculated. The center pixels are used to compute approximate line segments at 1040. At 1050, the vertical line segments are rotated -90°, putting them back in their original position as horizontal line segments. The approximate line segments are

mapped to world coordinates using homography information created during the calibration process. The process proceeds to 1080.

[0093] As for detecting vertical line marks, it is similar to detecting horizontal line marks, except that there is no need to rotate the image. For example, at 1060, the process finds vertical line marks and computes approximate line segments at 1070. The approximate line segments are mapped to world coordinates using homography information created during the calibration process. The process proceeds to 1080.

[0094] The vertical and horizontal line segments which are mapped to world coordinates, are checked for intersecting pairs of a vertical line and a horizontal line (V-H) at 1080. In one embodiment, V-H pairs are shortlisted. Shortlisted V-H pairs are those with the shortest distance between vertical and horizontal line segments which is below a predefined distance threshold d_{thd} . The offset d_{thd} , which is typically equal to the thickness of the line marks, is considered to allow small gaps between V-H line segments due to pixel rounding off in edge detection and in line fitting techniques.

[0095] The inner product of each vertical line segment with each horizontal line segment of the shortlisted V-H pairs is calculated according to equation 7 below:

$$VHAngle = \arccos(V.H / (|V| |H|)) * 180 / \pi \quad \text{Equation (7).}$$

To accommodate slots having different orientations with respect to the horizontal boundary, the angle of the V-H pair is not limited to 90°.

[0096] At 1090, a valid corner point pair (2 adjacent V-H pairs) is determined from the shortlisted V-H pairs. For example, a valid corner point pair, which includes 2 adjacent V-H pairs, is determined from the shortlisted V-H pairs. This is achieved by computing a confidence score which evaluates the geometric relationship between the set of two corners. The confidence score is based on the various characteristics of a parking slot. In one embodiment, the confidence score is based on the following characteristics: a) the width of

what a parking slot should be within a predefined width or range; b) the vertical lines of two V-H pairs should be parallel; and c) the supplementary angle property. The confidence score is calculated based on equations 8a, 8b, 8c, and 8d.

[0097] Regarding equation 8a, it is as follows:

$$C_w = 1 - (abs(Width_R - Width_A)/Width_R) \quad \text{Equation (8a),}$$

where

C_w is the confidence score based on width,

$Width_R$ is the reference width, and

$Width_A$ is the actual width between two reference corners.

[0098] Regarding equation 8b, it is as follows:

$$C_v = 1 - (abs(VAngle_L - VAngle_R)/90) \quad \text{Equation (8b),}$$

where

C_v is the confidence score based on the angles of two vertical lines,

$VAngle_L$ is the angle of the vertical line forming the left corner, and

$VAngle_R$ is the angle of the vertical line forming the right corner.

[0099] Equation 8c is as follows:

$$C_{hv} = 1 - (abs((VHAngle_L + VHAngle_R) - 180)/180) \quad \text{Equation (8c),}$$

where

C_{hv} is the confidence score based on angles of the V-H pair,

$VHAngle_L$ is the angle of the left V-H pair corner, and

$VHAngle_R$ is the angle of the right V-H pair corner.

The VH Angles are calculated according to equation 7.

[00100] As for equation 8d, it is as follows:

$$C = C_w * C_v * C_{hv} \quad \text{Equation (8d)}$$

where

C is the overall confidence score based on C_w , C_v , and C_{hv} .

[00101] The V-H pair with the highest confidence score is determined as the valid corner point pair. In the case where there are no corners, such as straight lines, equation 8c can be ignored.

[00102] Fig. 10b depicts an example of a parking slot detection 1001. A grayscale image 1011 is provided for slot detection analysis. The horizontal lines and vertical line marks in the image are detected. For example, image 1021 shows the vertical line marks while image 1022 shows horizontal line marks. The center pixels of the vertical line marks are shown in image 1031 and the center pixels of the horizontal line marks are shown in image 1032. Based on the combination of V-H pairs, a valid corner point pair is determined, as shown in images 1041 and 1042.

[00103] As described, parking slot markings detection is performed after occupancy check. In other cases, the parking slot markings detection may be performed before occupancy check. Other configurations of various steps of parking slot detection and validation may also be useful.

[00104] An experiment comparing the performance of the present autonomous parking technique using HGMS and techniques using a you-only-look-once (YOLO) algorithm and a pixel gradient algorithm was conducted. Two YOLO techniques were tested, one being a default YOLO technique and the other was custom trained using transfer learning with a training dataset of 400 images. The comparison was performed with an Intel Core i7-7700HQ CPU @ 2.80GHz, 8 core processor.

[00105] A test dataset of 623 different parking slots across different locations and at different times of the day was collected. Of the 632 parking slots, 402 were free slots and 221 were occupied slots. Among these slots, 65% are outdoor slots and about 10% of the slots have wet floors. Each slot data is a set of two successive image frames recorded at

20fps, with both image frames containing parking slot markings of a parking slot. The test dataset is different from the training data set

[00106] YOLO and pixel gradient are single frame techniques. As such, the first image frame of the parking slots was used. For the HGMS technique, both image frames were used.

[00107] Fig. 11a shows a comparison of occupancy check of the different techniques based on different input images. First, second, third, and fourth input images 1110, 1120, 1130, and 1140 are compared. The first and second input images include vehicles with different colors, the third input image is an unoccupied slot with shadows, and the fourth input image is an occupied space with traffic cones as obstacles. The masks of obstacles detected using a pixel gradient-based technique are depicted in the second column 1121. The third column 1131 shows rectangular boxes bounding objects detected using YOLO-based custom trained technique while the last column 1141 presents masks of obstacles computed by the HGMS.

[00108] As shown, the pixel gradient-based technique detected the vehicles but failed to detect an empty slot. Shadows and floor gradient affected the pixel gradient-based system as it is entirely based on gradient variations. In addition, pixel gradient-based system failed to detect obstacles other than vehicles.

[00109] The default YOLO-based technique (not shown) failed to detect unknown objects such as cones. The custom trained YOLO-based system can detect different objects with which it has been trained to detect and worked well with shadows. However, with new types of objects with which the system has not been trained, additional data collection, annotation and custom training are required. Unlike the other techniques, the masks of obstacles generated by the HGMS successfully detected different types of obstacles and accurately determine empty slots.

[00110] Fig. 11b shows a table of scores for performance indicators for the different types of systems related to detecting a free parking slot. The performance indicators include precision (P), recall (R), and the F1 (or accuracy) score, which is equal to $2*(P*R)/(P+R)$. Based on the comparison analysis, it is observed that the pixel gradient-based system, although simple and fast, has limitations in handling situations with varying intensities. As for the YOLO-based custom trained system, it shows robust performance in scenarios having varying intensities with known object types. However, the performance is poor when unknown obstacles are present and is the slowest among the systems compared. In other words, the YOLO-based system is highly dependent on the quality of and variations in the dataset used for training.

[00111] The HGMS technique is able to process an image frame in 45 msec, which is near real-time. As for the YOLO-based technique, in addition to the dependency of training, processing a frame takes the longest time among the techniques compared. As shown, the HGMS technique achieves high performance ratings. Furthermore, the HGMS technique is not dependent on color and shape of the obstacles and exhibits superior occupancy detection for different textures and shadows in parking slots. For example, the HGMS technique is able to detect objects of different sizes such as cars, trees, traffic cones and others. This is primarily due to the underlying principles of motion estimation which are immune to these variations. With an F1 score of 97%, the HGMS performance is at par with the YOLO techniques with respect to known obstacles, shadows, dirt, wet surfaces, but at a rate which is near real-time.

[00112] The inventive concept of the present disclosure may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. The foregoing embodiments, therefore, are to be considered in all respects illustrative rather than limiting the invention described herein. Scope of the invention is thus indicated by the

appended claims, rather than by the foregoing description, and all changes that come within the meaning and range of equivalency of the claims are intended to be embraced therein.