AUGMENTED REALITY-BASED GUIDANCE FOR INDOOR PARKED-CAR SEARCH USING OMNI-VISION TECHNIQUES†

[1]*Ya-Jui Hsieh* (謝亞叡) *and* [2]*Wen-Hsiang Tsai* (蔡文祥)

[1]Institute of Computer Science and Engineering
[2]Department of Computer Science
National Chiao Tung University, Hsinchu, Taiwan
Emails: eric66033@hotmail.com.tw, whtsai@cis.nctu.edu.tw

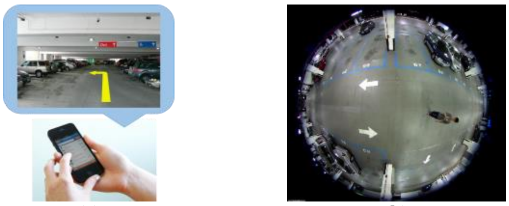
ABSTRACT

Many people have the experience of getting lost in large parking lots when trying to find their parked cars. To solve this problem, an augmented reality (AR)-based automatic guidance system is proposed to help a user to find his/her parked car quickly. With multiple fisheye cameras affixed to the ceiling of the parking lot for image acquisition, the proposed system includes four major functions: user detection, user tracking, path planning, and AR-based guidance for parked-car search. To detect a user entering the parking lot, a dynamic background construction method is proposed to segment the foreground regions in a fisheye image and a dynamic thresholding method is proposed to refine the user region in the segmented regions. To track the user continuously in a large parking lot, a multiple-camera handoff method is proposed, which transfers monitoring responsibility among the cameras according to a criterion of minimizing the distance between the user's location and the image center. Also proposed is the use of Kalman filtering for generating a smooth human track shown on the top-view parking-lot map. For path planning, a method based on the Dijsktra algorithm which yields a shortest guidance path from the entrance to the user's parked car. To guide the user, an AR-based guidance method is proposed, which updates and overlays a guidance arrow on the screen of the user-held mobile device by utilizing the reading of the electronic compass built in the mobile device as well as the location of the next guidance path node. Good experimental results are also included to show the feasibility of the proposed system.

Keywords: *augmented reality, parked-car search, automatic guidance, fisheye camera, user detection and tracking, path planning, mobile device, electronic compass.*

1. INTRODUCTION

Finding a parked car before leaving the parking lot is often a problem, especially when the parking lot is large or when the driver forgets his/her parked-car location or the parking-space number. A feasible solution is to design an augmented reality (AR)-based guidance system for parked-car search, which has the functions of both *remembering the location of a parked car* and *guiding the user to find the car*. It is aimed in this study to design such an AR-based guidance system for use by a user with a mobile device (a smart phone or pad). Such a system requires the core technique of *user positioning* which we implement by analyzing the images captured from the fisheye cameras installed on the ceiling of the parking lot, and detecting human activities in the environment by a mixture of image and signal processing techniques. Also, the system uses AR techniques to overlay the car-search guidance information in the form of a *guiding arrow* on the parking-lot image captured by the camera built in the user-held mobile device. By following the augmented guiding arrow on the mobile-device screen, the user can reach his/her parked car. These functions are illustrated in Fig. 1.



(a) (b)

Fig. 1: Illustration of proposed AR-based guidance system for parked-car search. (a) An illustration. (b) An image of a parking lot used in proposed system.

About indoor guidance systems, more and more related techniques have been proposed recently. Based on sensor-based techniques, Chittaro et al. [1] proposed an indoor guidance system using the radio frequency identification (RFID) technique, which determines a user's position and gives evacuation instructions on mobile devices based on interactive location-aware 3D models of the building. Ozdenizci et al. [2] proposed an indoor guidance system using the near field communication (NFC) technique, which enables an easy data transfer process just by touching tags spread over a building or a complex with a smart phone. Mulloni et al. [3] proposed an AR interface to support indoor guidance by wireless networks, which combines activity-based instructions with sparse 3D localization at selected information points in the building.

Based on image-based techniques, marker-based positioning using camera phones is the most commonly-used technique in various indoor guidance

† This work was supported financially by NSC project No. 102-2221-E-009-106-.

[2] Wen-Hsiang Tsai is also with the Dept. of Information Communication, Asia University, Taichung, Taiwan 41354.

systems. Barberis et al. [4] proposed an indoor guidance system using a set of natural image features and visual markers for user positioning. Moller et al. [5] adopted a vision-based approach by image matching using extracted environment features to implement an indoor guidance system. They also combined virtual reality (VR) and AR elements with indicators to help communication and ensure localization accuracy.

About related guidance applications, Miyashita et al. [6] designed an AR-based museum guide system which combines markerless tracking, hybrid tracking, rotation sensors, and an ultra-mobile-PC. Kim and Jun [7] proposed a system which automatically recognizes a location from image sequences taken of indoor environments, and realizes AR by overlaying the user's view with location information. Kim and Dey [8] proposed the concept of an in-vehicle guidance display system that displays guidance information directly onto the vehicle's windshield and overlays it on the driver's view of the actual road.

In this study, we try to develop an AR-based guidance system for car search for use by a driver before leaving a parking lot, which has the following capabilities: 1) computing the positions of the user and the parked car automatically and show them on a map; 2) planning a shortest guidance path to lead the user to his/her parked car; 3) displaying the planned path on the screen of the user-held mobile device in an AR manner (i.e., with a guiding arrow drawn on the mobile-device screen); 4) updating the guiding arrow on the mobile-device screen dynamically, by which the user can inspect and walk accordingly to reach the destination space where his/her parked car is located.

The remainder of this paper is organized in the following way. The configuration of the proposed system and the system processes are described in Sec. 2. A process of "learning" the parking-lot environment is presented in Sec. 3. A new method for user detection and tracking is proposed in Sec. 4. A path planning process for generating a shortest path from the user to the parked car as well as an AR-based method for rendering and updating the guidance arrow overlaid on the mobile-device screen are described in Sec. 5. Some experimental results are presented in Sec. 6, followed by some conclusions in the last section.

2. SYSTEM DESIGN AND PROCESSES

2.1 System Design

As shown in Fig. 2, at first we affix a number of fisheye cameras on the ceiling of the parking lot where the proposed system is to be applied. The cameras are used to acquire images of the parking-lot environment, by which a user entering the parking lot can be detected and tracked for user positioning. Also, the system is constructed to be of a client-server structure. The server, which is located at a remote site and connected to the fisheye cameras through an Ethernet, is designed to conduct the works of planning a path from the entrance location to the user's parked car and sending the guidance information to the client which is run on the user-held mobile device. When a user enters the parking lot, the client is connected to the server through the network, and starts to receive the guidance information from the server and display the augmented information on the image shown on the mobile device.

Fig. 1: An illustration of system design and processes.

2.2 Learning Process

The software operations of the proposed system include two processes – *learning* and *guidance*. As shown in Fig. 3, the first goal of the learning process is *learning of the environment*, meaning acquisition of the information of the parking lot, such as the parking-space positions and the fisheye-camera positions which are data required for the guidance process. The second goal is *learning for path planning*, meaning creation of path nodes (like path crossings) for use by the path planning method based on the Dijkstra algorithm. The third goal is *learning for user localization*, meaning acquisition of the magnetic-field data in the parking lot which is used for computing the user orientation and conducting space mapping between the acquired image space and the real-world space is shown as a parking-lot map in this study.

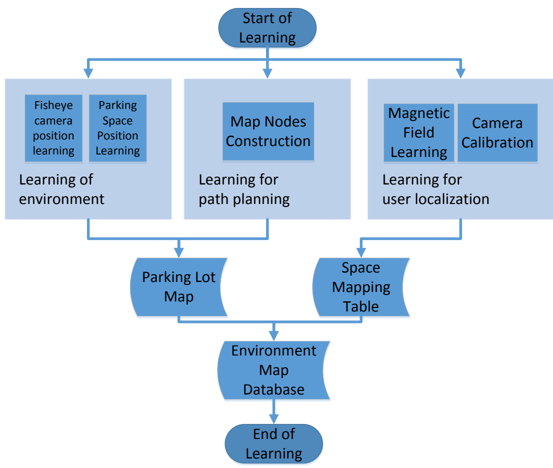


Fig. 3: A framework of proposed learning process.

2.3 Guidance Process

Three main tasks are conducted in the guidance process, namely, (1) path planning, (2) detection and tracking of the user, and (3) AR-based guidance for parked-car search. The roles of these three tasks in the entire guidance process are shown in Fig. 4. The first task, path planning, is to generate a suitable shortest path to guide the user from his/her location to the destination after the position of the entrance and the location of the user's parked car are known. The second task is the detection and tracking of a user. When a fisheye image is acquired by a camera, the system detects the user in the image and track him/her continually. The third task is AR-based guidance for parked-car search which aims to analyze the user's orientation, create a proper and dynamic guidance arrow, and overlay it on the image captured by the user's phone camera in an AR fachion.

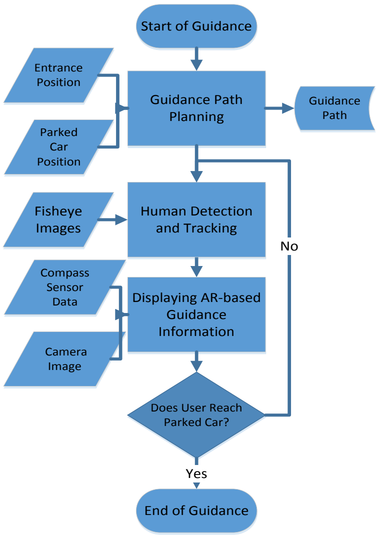


Fig. 4: A framework of guidance process.

3. LEARNING OF ENVIRONMENT

3.1 System Learning

Four types of parking-lot environment parameters are measured and saved into the PC server in the proposed learning process. The first is the identical heights H of the fisheye cameras on the ceiling. With this parameter H, a space-mapping lookup table is created to transform from the fisheye image into the top-view image which is more agreeable for human inspection and can be used to locate a user on the *environment map* easily.

The second information needed is the locations of the fisheye cameras and the parking spaces, which are required data for conducting transformations related to the *global coordinate system* (GCS) built on the parking lot environment. The human tracking process needs such information to locate the user precisely. Also, the positions of the parking spaces are used for locating the user's parked car as the guidance destination. The third information needed is the magnetic-field parameter of the environment, which is used for user orientation detection. The user orientation can be calculated by comparing the information obtained by the mobile-device sensor and the magnetic-field information of the parking lot environment. Accordingly, an AR-based guidance arrow can be created to point out the right direction for the user to follow on the mobile-device screen. The last information is the locations of the *map nodes* which we construct for path planning in the parking lot. With these map nodes, a guidance path can be generated to guide the user from his/her current position to the destination which is his/her parked car. According to these types of information mentioned above, the previously-mentioned environment map can be constructed. An example is shown in Fig. 5.

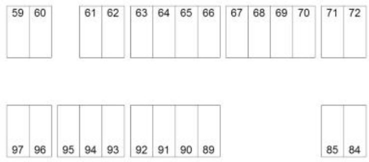


Fig. 5: An illustration of constructed environment map.

3.2 Review of an Adopted Space Mapping Method

To detect users in the parking lot, we need top-view images of the parking lot. For this, we conduct image-space mapping according to Chen and Tsai [9] from the *fisheye-image coordinate system* (*FCS*) to the *map coordinate system* (*MCS*), by which we can obtain a top-view image of the parking-lot environment from an input fisheye image. The entire process of space mapping can be illustrated by Fig. 6. Firstly, we take an image of a *calibration cylinder* with a grid pattern (shown on the right-hand of Fig. 6), and find the corner points of it in the image. Secondly, a *space-mapping table* T_F is created according to a geometric model. Finally, a transformation from the fisheye image into the top-view image can be derived by looking up the table T_F.

More specifically, a calibration cylinder as just mentioned is constructed at first, which is shown in Fig. 7(a). Inside it is attached a chessboard-like grid pattern whose squares are of the size of 2×2 cm^2. The circles formed by the grids on the bottom are designed to have increasing radii from inside to outside. Secondly, the cylinder is placed under each of the fisheye cameras, and an image of it is taken with the fisheye camera, called a *space-mapping image*. An example is shown in Fig. 7(b). Thirdly, a corner detector is applied to obtain the corner points of the squares in the image, as shown as the yellow points in Fig. 7(c). Subsequently, a space-mapping function, described as a table T_F, is derived, by which we can compute the corresponding location on the top-view image of each corner point in the space-mapping image.

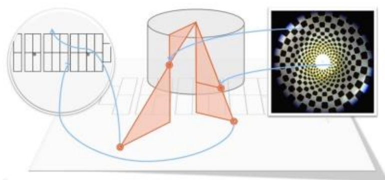
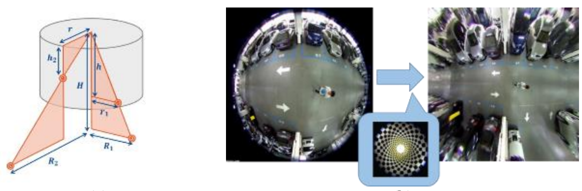


Fig. 6: An illustration of image space mapping between fisheye image and top-view image.


(a) (b) (c)
Fig. 7: An illustration of the space-mapping cylinder.

In more detail, an illustration of the space-mapping function is shown in Fig. 8(a) where the value r and h are the radius and height of the cylinder, respectively; H is the height of the fisheye camera with respect to the floor; r_1 is the distance of a corner point P_{cb} on the bottom of the cylinder to the center of the cylinder; and h_2 is the distance from a corner point P_{cs} on the "vertical" side of the cylinder to the top of the cylinder, which is an input known value. The output variables are the two distances R_1 and R_2 of the points P_{cb}' and P_{cs}' in the top-view image corresponding to P_{cb} and P_{cs}, respectively, as shown in the figure. According to the similar-triangle principle, we have $R_1/r_1 = H/h$ and $R_2/r = H/h_2$, leading to the formulas for computing R_1 and R_2: $R_1 = (H/h)\times r_1$ and $R_2 = (H/h_2)\times r$. For non-corner points, the values of R_1 and R_2 are computed by interpolation. As a result, every point in the space-mapping image is mapped to a point in the top-view image using the formulas for R_1 and R_2. This mapping can be represented as a table which is just Table T_F mentioned previously. Subsequently, given a fisheye image, we can then transform it into a top-view image conveniently by table-lookup using T_F. An experimental result of conducting such a mapping is shown in Fig. 8(b).



(a) (b)

Fig. 8: Mapping from fisheye images to top-view images. (a) An illustration of notations involved in the mapping. (b) An experimental result.

It is not difficult to figure out that the previously-described mapping is reversible so that we may construct a *backward space-mapping table* T_R from T_F, which can then be used to map a top-view image back into a fisheye image by table-lookup.

4. HUMAN DETECTION AND TRACKING BY DOWN-LOOKING FISHEYE CAMERAS

In this study, the human detection and tracking function is not only the core of the guidance process but also the key of the augmented reality process. In order to achieve these goals, three methods are proposed in this study to implement this function, namely: (1) human detection by image processing techniques; (2) human orientation computation by use of the reading of the electronic compass built in the user-held mobile device; and (3) human tracking by camera handoff and Kalman filtering techniques.

4.1 Human Detection by Image Processing

To detect a user in an input fisheye image, at first the user's body part is extracted. For this, a background image is constructed dynamically as described in Algorithm 1. Then, when the user enters the environment in the guidance process, he/she is identified by the server a new-coming user using the unique ID number of his/her mobile device which is sent to the server. Next, his/her body is found from the acquired fisheye-camera image by a process of foreground-region detection, including background subtraction, dynamic thresholding, and region growing. The dynamic thresholding scheme is based on the assumption that the size of the same user' region will not change rapidly between two adjacent moments. Extraction of the user's body region by dynamic thresholding is introduced in Algorithm 2. In addition, the user's foot point in the found body region can be detected according to an optical property of the fisheye camera — in an image acquired with a downward-looking fisheye camera, a space line perpendicular to the ground appears as a radial line going through the fisheye image center. Accordingly, the axis of the user's body will go through the image center. Therefore, the user's foot point may be found to be the image point in the detected body region nearest to the image center. Finally, the user's foot point is transformed into the MCS by the forward space-mapping process mentioned previously.

Algorithm 1. Background Image Construction and Subtraction

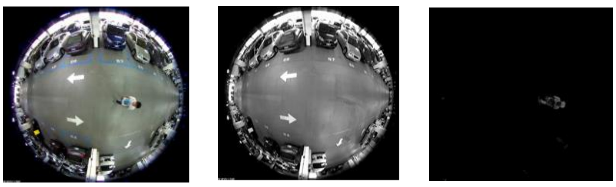
Input: an image I captured with a fisheye camera.

Output: a foreground image F constructed by background subtraction.

Steps.

1. Determine if a background image B_0 is available or not: if yes, go to Step 2; otherwise, go to Step 3.
2. Construct a new background B_N with the following formula, where α is a parameter which can influence the rate of background updating; and go to Step 4:
$$B_N = (1-\alpha)\times B_0 + \alpha\times I \qquad (1)$$
where I is the input image.
3. Assign the first fisheye image I to be the first background B_N.
4. Construct a foreground image F by conducting frame differencing on the input fisheye image I and the background image B_N constructed in the previous steps.
5. Replace the old background B_0 with the new background B_N for the next cycle.

An experimental result yielded by the above algorithm is shown in Fig. 9.



(a) (b) (c)

Fig. 9: An experimental result yielded by Algorithm 1. (a) An input image captured with a fisheye camera. (b) A background image constructed by Steps 2 and 3 of Algorithm 1. (c) A foreground image constructed by this algorithm.

Algorithm 2. Extraction of the User's Body Region by Dynamic Thresholding

Input: The foreground image F of the fisheye image.

Output: The user's region R_u in the foreground image F.

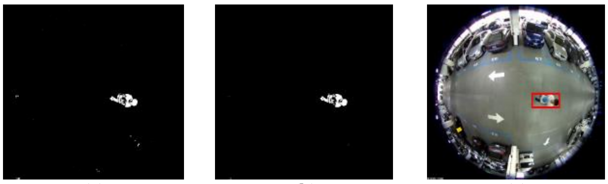
Steps.

1. Apply an initial threshold value T_D on the foreground image F by the following steps:

(1) set $F(u, v) = 1$, if $|F(u, v)| > T_D$;;
(2) set $F(u, v) = 0$, otherwise,
where $F(u, v)$ denotes a pixel value in F at (u, v).

2. Remove all small connected components in F to eliminate noise which results from the above step.
3. Find the remaining connected components in F as the desired foreground regions R_1, R_2, …, R_n and circumscribe them together with a red bounding box.
4. Compare the bounding box size with the previous one which is constructed in the last cycle:
   (a) if the size is much bigger than the previous one, go back to Step 1 and increment threshold T_D;
   (b) if the size is much smaller than the previous one, go back to Step 1 and decrement threshold T_D;
   (c) otherwise, continue.
5. Regard the final region circumscribed by the red bounding box as the user's region.

An experimental result yielded by above algorithm is shown in Fig. 10.



(a) (b) (c)
Fig. 10: An experimental result yielded by Algorithm 2. (a) Foreground image after thresholding. (b) Foreground image after removing small components. (c) Resulting image after circumscribing the user's regions.

4.2 Human Orientation Computation by Electronic Compass

The computation of the human orientation plays an important role in the augmented reality process. The guidance arrow is displayed on the user's mobile-device screen to guide the user correctly only if the computed human orientation is correct. To detect the user's orientation, we use the reading values of the electronic compass which is the orientation sensor built in the mobile device. Firstly, we assume that the user holds the mobile device vertically as shown in Fig. 11(a) because of the design of the application interface in this system. Accordingly, we can establish the user's orientation just by the azimuth angle measured by the orientation sensor. However, the orientation sensor is not always very precise. Fortunately, based on the fact that the paths in the parking lot have only four main directions as shown in Fig. 11(b) which are only perpendicular or parallel to each other, we can assume that the user only faces these four orientations when he/she is walking on the paths. Therefore, we propose a knowledge-based orientation decision and correction method to deal with the accuracy problem which is introduced in Algorithm 3.

Algorithm 3. Knowledge-based Orientation Decision and Correction.

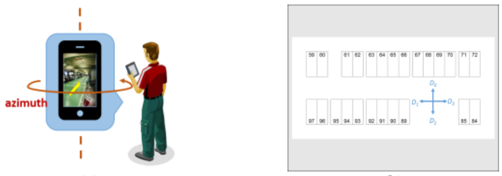
Input: an azimuth angle A measured by an electronic compass.

Output: the user's orientation O in one of the four major directions.

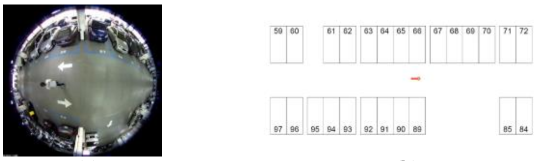
Steps.

1. Initialize the azimuth angles of the four major directions, D_0 through D_3, in the environment, which are recorded in the learning stage.
2. Divide the angles which are from 0 to 360° into four ranges, $D_0' = (D_0 - 45°, D_0 + 45°)$, $D_1' = (D_1 - 45°, D_1 + 45°)$, $D_2' = (D_2 - 45°, D_2 + 45°)$, and $D_3' = (D_3 - 45°, D_3 + 45°)$.
3. Determine which range D_i' that the azimuth angle A belongs to where $i = 0 \sim 3$.
4. Decide and correct the user's orientation O to be the corresponding direction D_i.

An experimental result yielded by the above algorithm is shown in Fig. 12.



(a) (b)
Fig. 11: Human orientation computation. (a) An illustration of the scenario in the guidance process. (b) An illustration of four major directions on the environment map.



(a) (b)
Fig. 12: An illustration of user orientation decision by Algorithm 3.

4.3 Human Tracking by Uses of Camera Handoff and Kalman Filtering Techniques

The objective of human tracking is to identify the same person in the consecutive video frames of a video captured by a fisheye camera so that the guidance system can localize the same user and send proper guidance information to the user continuously. At first, the foreground regions in a captured image are extracted, and the user's region is enclosed with a bounding box, as described previously. Also, the user's location is transformed into proper coordinates for showing on the environment map. The tracking process can just be done by finding the location point nearest to the user's location of the last moment. However, the resulting trajectory is *not smooth* because of the existence of noise or imprecise user localization results. Also, the user can be tracked only by a single fisheye camera; cross-camera tracking should be carried out to achieve complete tracking in the entire parking lot space.

To deal with the above-mentioned problems, a camera handoff method is proposed which is introduced in Algorithm 4. The camera handoff process can track a user continuously when the user crosses the field of a fisheye-camera's view to another. Also, we adopt the Kalman filter to reduce the influence of noise, localize the user with higher precision, and improve the user-track continuity resulting from human detection. The details of using the Kalman filter to improve the user's trajectory smoothness in this study for more convenient inspection are described in Algorithm 5.

Algorithm 4. Camera handoff process.

Input: multiple user regions U_i in omni-images I_i captured by fisheye cameras C_i, respectively, where $i = 1, 2, \ldots, N$ with N being the number of fisheye cameras in the parking lot.

Output: the user's location P on the environment map which is localized by the image captured from the nearest fisheye camera C_n.

Steps.

1. Initialize the azimuth angles of the four major directions, D_0 through D_3, in the environment, which are recorded in the learning stage.
2. Select the fisheye camera C which is the nearest one at the entrance, and call it the *nearest camera* and denote it by C_N.
3. Take the fisheye camera C which is used in the last user-tracking cycle as the nearest camera C_N.
4. Compute the user's location P on the environment map corresponding to the user region U which appears in the image captured by the nearest camera C_N.
5. Find the new fisheye camera C_n which has the smallest distance to the user's location P on the environment map; and if C_n is not the same as the nearest camera C_N, replace C_N by C_n and go back to Step 4; otherwise, continue.
6. Regard the point P as the user's location on the map, record the selected nearest camera C_N for use in the next tracking cycle, and go to Step 3.

Algorithm 5. Using Kalman Filter for Keeping Track Continuity.

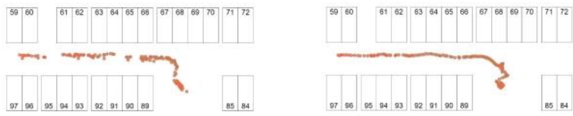
Input: the user's location P_m on the environment map which is the measurement obtained by human detection process.

Output: the user's location P_k on the environment map which is calculated by the Kalman filter..

Steps.

1. Initialize related parameters, including (1) a transition matrix T used to predict the next state of user's location; (2) a value N_p of *process noise* which is produced by the error prediction process and assumed to be a random value given by a normal distribution; (3) a *measurement noise* N_m which is produced by the measurement obtained by image processing and also assumed to be a random value given by a normal distribution; (4) an initial prediction error E_p used to determine the error of the predicted location; and (5) the initial user's location P_k.
2. Use the Kalman filter to predict the user's location P_p and compute a prediction error E_p in the following way:
$$P_p = T \times P_K$$
$$E_p = T \times E_P + N_p$$
where P_K is the user's location estimated in the last cycle and E_P is the prediction error updated in the last cycle.
3. Compute the Kalman gain K_g with the prediction error E_p and the measurement noise N_m in the following way:
$$K_g{}^2 = E_p{}^2 \big/ (E_p{}^2 + N_m{}^2)$$
4. Compute the user's location P_k in terms of the predicted user's location P_p obtained in the prediction stage described in Step 2, the measurement location P_m, and the Kalman gain K_g in the following way:
$$P_k = P_p + K_g \times (P_m - P_p)\,.$$
5. Assign the estimated user's location P_k computed in the last step to P_K, update the prediction error E_P with the Kalman gain K_g in the following way, and go back to Step 2 for the next cycle:
$$E_P = \sqrt{(1 - K_g) \times E_p{}^2}\,.$$
6. Regard the point P as the user's location on the map, record the selected nearest camera C_N for use in the next tracking cycle, and go to Step 3.

An experimental result yielded by above algorithm is shown in Fig. 13. We can easily see that the user-track continuity is much better than before.



(a) (b)

Fig. 13: An experimental result of using the Kalman filter (a) The trajectory output of the human detection process. (b) The trajectory improved by the use of the Kalman filter.

5. PATH PLANNING AND AR-BASED GUIDANCE FOR PARKED-CAR SEARCH

5.1 Path Planning for Parked Car Guidance

Path planning is necessary for user guidance to the destination which is his/her parked car, especially when the user forgets where his/her parked car is. In the path planning process, the first issue is finding the start point of a planned path. In the previous sections, we have already described the method for user localization. Therefore, the detected location of the user at the entrance of the parking lot can be used as the start point of the guidance path. The second issue is finding the destination point of the planned path. Because the system proposed by Chen and Tsai [9], which has the functionality of guiding the user to an empty parking space, is combined into our system, the location of the parking space where his/her car was parked has already been known and kept in the system server when Chen and Tsai's system was performed in advance. That is, the known parked-car location can be used as the destination point of the planned path.

Finally, the Dijkstra algorithm is employed to solve the shortest path planning problem in our study. The map nodes used for the Dijkstra algorithm have been created in the learning stage. They are spread on the entrance, road, and parking space areas in the parking lot. Each map node has some information recorded to conduct the path planning process. The first kind of information is the location on the environment map which is used to determine which node the user reaches now. The second kind of information is a list of adjacent nodes used to decide where the user can go next on the current node. Also, if the map node is right on a parking space, it is necessary to record the parking-space number to determine which node should be the user's destination. After the above-mentioned process, a planned path like that shown in Fig. 14 can be constructed.

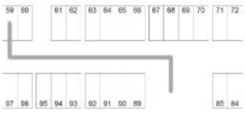


Fig. 14: An experimental result of path planning process. The path is from the entrance to the user's parked car which is on the 59-th parking space.

5.2 Displaying guidance arrow on mobile-device screen

After the user's location and orientation as well as the planned path are known, a proper guidance arrow can be drawn and displayed on the mobile-device screen. The information of the user's location can determine which node on the planned path the user has reached. Accordingly, the planned path can decide step by step which node will be the next destination for the user to reach. Finally, the information of the user's orientation can decide which direction the user should turn to for the next node and a proper arrow can be drawn accordingly on the mobile-device screen for the user to follow. Algorithm 6 describes how a proper guidance arrow is drawn according to these three kinds of information.

Algorithm 6. Displaying a proper guidance arrow on a mobile-device screen.

Input: the current user's location L, the current user's orientation O, and a planned path P.

Output: a proper guidance arrow A drawn on mobile device screen.

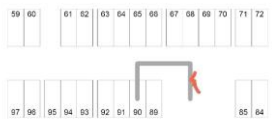
Steps.

1. Determine which map node N_c the user has reached by use of the current user's location L.
2. Decide which node N_n is the next destination that should be reached according to the planned path P and the current node N_c.
3. Select a proper guidance arrow A according to the orientation D from the current node to the next and the current user's orientation O in the following way:
   (a) if $(D - O) = 0^o$, select a *forward arrow*;
   (b) if $(D - O) = -90^o$ or 270^o, select a *leftward arrow*;
   (c) if $(D - O) = 90^o$ or -270_o, select a *rightward arrow*.
4. If the user reaches the node N_n, assign N_n as the current node N_c and go back to Step 2; otherwise, display the guidance arrow selected in Step 3.

An experimental result yielded by above algorithm is shown in Fig. 14.

6. EXPERIMENTAL RESULTS

Some experimental results of applying the proposed system are shown here. The experimental environment is a parking lot in National Chiao Tung University. There are 25 parking spaces there, and four down-looking fisheye cameras are installed on the ceiling, mainly above the main paths in the parking lot.

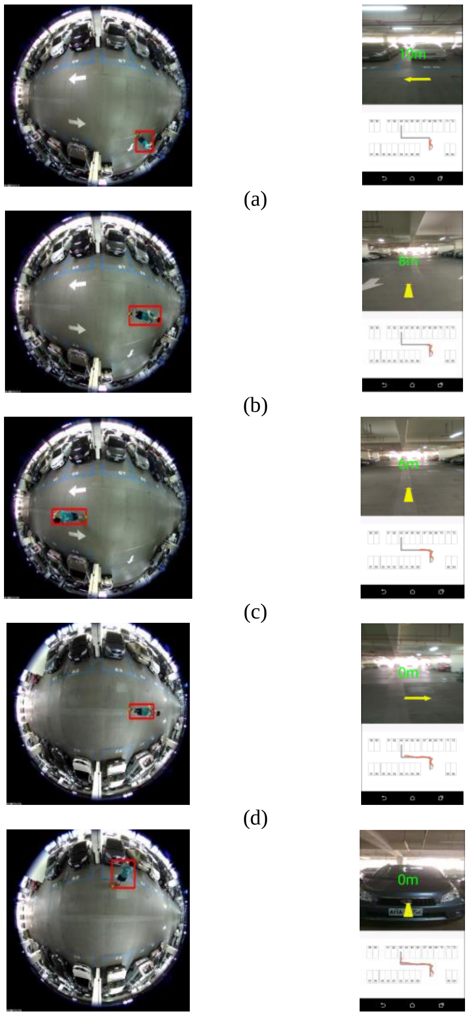


(a)

(b)

Fig. 14: An experimental result of the augmented reality-based guidance. (a) is a map of the parking lot, where the gray line is a planned path and the red points represent the user's trajectory. (b) is an AR-based guidance image displayed on the user's mobile device.

The left-side images of Fig. 15(a) through (e) are the fisheye images used for user detection. And the right-side images are the scene of the corresponding locations with the trajectories of the detected user on the environment map and the planned path also shown. These results show that a user entered the parking lot, and moved toward his/her parked car which is on the 63-th parking space. From these images, it can be seen that the proposed method works effectively.



(a)

(b)

(c)

(d)

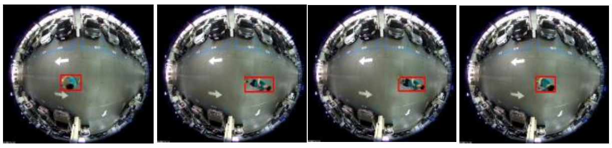
(e)

Fig. 15: A series of experimental results of the proposed AR-based guidance system. The left-side image in each of (a) through (e) is the result of human detection. In the respective right-side image, the green number shows the remaining distance to the destination, the yellow arrow points toward the correct direction for the user to move forward, and the red points show the user's trajectory.

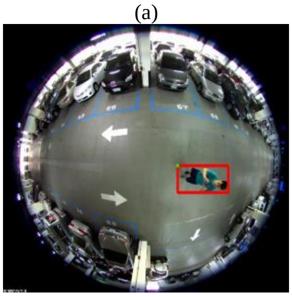
A series of experiments have also been conducted to compute the human detection rate, and the result is shown in Table 1. One hundred frames in a video clip were tested and only one error occurred, as shown in Fig. 16. So, the correct detection rate is 99%. It was found that the only error in the 100 tested frames came from the user's shadow which influences the foot point detection.

Table 1: An experiment of human detection rate.

Number of Test Frames (N)	Number of Correct Detections (n)	Correct Detection Rate (n/N)
100	99	99 / 100 = 99%



(a)



(b)

Fig. 16: A series of experimental results for computing the human detection rate. (a) Some correct detection results. (b) An erroneous detection result owing to the influence of human shadow.

7. CONCLUSIONS

An AR-based guidance system for parked-car search in parking lots using multiple down-looking fisheye-cameras has been proposed in this study. Many techniques have been proposed or adopted to implement the system, which are summarized in the following.

(1) Skillful schemes including dynamic background construction and dynamic thresholding for human detection and localization are designed and applied effectively for finding the precise position of the user in fisheye-camera images of parking lots.

(2) A human tracking method based on the uses of Kalman filtering and computer vision techniques has been proposed for generating a smooth human track shown on a top-view parking-lot map, helping the user to locate himself/herself as well as to avoid getting lost in the parking lot.

(3) A method of knowledge-based user-orientation decision and correction by the use of the sensors built in smart phones has been proposed, which helps drawing proper AR-based guidance arrows to show on the mobile-device screen for the user to inspect and follow toward the target of the .parked car.

(4) A multiple-camera handoff method has been proposed for user tracking in a large-size parking lot using multiple fisheye cameras attached on the ceiling of the parking lot.

(5) A method for planning a guidance path for parked car search has been proposed, by which the system can plan a suitable minimum path from the current user's position to the his/her parked car for the user to follow efficiently.

(6) A method for rendering the AR-based guidance arrow on an image captured with the mobile-device camera is proposed, and by following the guiding arrow shown on the mobile-device screen, the user can be guided to reach his/her parked car conveniently and easily.

(7) A technique for updating dynamically the guidance arrow shown on the user mobile-device screen when the user face toward the wrong direction has been proposed, which utilizes the reading values of the electronic compass built in the user mobile device under the assumption that the walk ways of the parking lot are composed of perpendicular path segments.

Good experimental results reveal the feasibility of the proposed system.

REFERENCES

[1] L. Chittaro and D. Nadalutti, "Presenting evacuation instructions on mobile devices by means of location-aware 3D virtual environments," *Proceedings of International Conference on Human Computer Interaction with Mobile Devices and Services*, New York, USA, 2008, pp. 395-398.

[2] B. Ozdenizci., O. Kerem; V. Coskun and M. N. Aydin, "Development of an indoor navigation system using NFC technology " *Proceedings of 4th International Conference on Information and Computing (ICIC)*, Phuket Island, Thailand 2011, pp. 11-14.

[3] A. Mulloni, H. Seichter and D. Schmalstieg, "Handheld augmented reality indoor navigation with activity-based instructions," *Proceedings of International Conference on Human Computer Interaction with Mobile Devices and Services*, New York, USA, 2011, pp. 211-220.

[4] C. Barberis, A. Bottino; G. Malnati and P. Montuschi, "Experiencing indoor navigation on mobile devices," *IT Professional*, vol. 16, Issue 1, 2014, pp.50-57.

[5] A. Moller, M. Kranz, R. Huitl, S. Diewald and L. Roalter, "A Mobile Indoor Navigation System Interface Adapted to Vision-Based Localization," *Proceedings of International Conference on Mobile and Ubiquitous Multimedia*, New York, USA, 2012, Article No. 4, pp. 31-40.

[6] T. Miyashita, P. Meier, T. Tachikawa; S. Orlic, T. Eble, V. Scholz, A. Gapel, O. Gerl, S. Arnaudov and S. Lieberknecht, "An augmented reality museum guide," *Proceedings of IEEE International Symposium on Mixed and Augmented Reality(ISMAR)*, Cambridge, UK, 2008, pp. 103-106.

[7] J. Kim and H. Jun, "Vision-based location positioning using augmented reality for indoor navigation," *Proceedings of IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, 2008, pp. 954-962.

[8] S. Kim and A. K. Dey, "Simulated augmented reality windshield display as a cognitive mapping aid for elder driver navigation," *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, New York, USA, 2009, pp. 133-142.

[9] J. Chen and W. H. Tsai, "Automatic guidance for indoor car parking using augmented reality and omni-vision techniques," *Proceedings of 2014 Conference on Computer Vision, Graphics and Image Processing*, Pintung, Taiwan, Aug. 2014.