유비쿼터스 주차관리를 위한 차량충돌 검증시스템☆

Car Collision Verification System for the Ubiquitous Parking Management

마테오 로미오* 양 현 호** 이 재 완***
Romeo Mark A. Mateo Hyunho Yang Jaewan Lee

요 약

WSN기반 주차관리 시스템에서 대부분의 연구는 주차장에서 사건을 통제하기 위해 무선 센서를 이용하지만, 주차장에서의 차량충돌에 대한 연구는 거의 수행되지 않았다. 시간에 따른 자세한 차량의 위치는 충돌 사건을 분석하는데 매우 중요하다. 본 연구는 주차장에서 차량 충돌사건을 감지하여 분석하고, 이를 차주에게 통보하는 충돌감지 방법을 제시한다. 차량의 위치 및 이동 방향을 감지하기 위해, 움직임 센서로부터의 정보를 활용하며, 빠른 OBB 교차 테스트를 사용하여 검증을 위한 객체를 추적한다. 성능평가 결과 위치추적 기법은 센서를 추가함에 따라 좀 더 정확함을 보였고, 제안한 OBB 충돌 테스트가 일반적인 OBB 교차테스트에 비해 속도가 향상됨을 나타내었다.

ABSTRACT

Most researches in WSN-based parking management system used wireless sensors to monitor the events in a car parking area. However, the problem of car collisions in car parks was not discussed by previous researches. The car position details over time are vital in analyzing a collision event. This paper proposes a collision verification method to detect and to analyze the collision event in the parking area, and then notifies car owners. The detection uses the information from motion sensors for comprehensive details of position and direction of a moving car, and the verification processes an object tracking technique with a fast OBB intersection test. The performance tests show that the location technique is more accurate with additional sensors and the OBB collision test is faster compared to a normal OBB intersection test.

☞ keyword : Parking management system(주차관리시스템), wireless sensor network(무선센서네트워크), collision detection(충돌감지), video tracking(비디오 추적)

1. Introduction

The current research studies in parking management system utilize the ubiquitous technologies to automate car monitoring and to provide smart processing of information in the wireless environment. Most studies implement wireless sensor devices in sensing the presence of cars and monitoring location of cars [1]. Relevant services like car locator system, parking negotiator and other applications in a parking system rely on the sensing methods to provide the basic information for parking system management. However, the previous studies did not include the problem of detecting and analyzing the collision event which is important for the management and car owners. An example of a situation is when a moving car, on its way out of the parking area, accidentally collided on a parked car. The owner of the parked car, which is not in the scene, cannot know that his or her car was hit by another car. In the case of moving cars, the car caused the collision is verified by the detailed position, speed and direction of each car. To solve

* 준 회 원 : 군산대학교 전자정보공학부 박사과정
rmmateo@kunsan.ac.kr
** 정 회 원 : 군산대학교 정보통신공학과 교수
hhyang@kunsan.ac.kr
*** 종신회원 : 군산대학교 정보통신공학과 교수
jwlee@kunsan.ac.kr (교신저자)

[2011/05/18 투고 - 2011/05/20 심사 - 2011/08/22 심사완료]
☆ This paper is supported by the Industry and University Consortium of Small and Medium Business Administration (SMBA) of 2010

these problems, the system should use a collision detector device embedded in the car or track the collision event using video recorders. This paper studies the collision detection in parking environment using wireless sensors and video tracking technique. There are three cases of collisions with the involvement of cars in a parking area which as follows:

- A moving car collides to another non-moving car
- A moving car collides to another moving car
- An unknown object (non-car) collides to a non-moving or moving car

In the first case, we know that the driver of the moving car has the fault of collision. The only problem is that the car caused the collision can avoid the responsibility because there is no system to identify the car and to detect the event. In the second case, we can know its either or both cars caused the collision. This is proved by the given details of the event like current position, speed, direction and others. To determine the details, each car must have identification and there must be a location technique to estimate the position of cars in the system. In the last case, it is impossible to identify the cause of the collision except if there are video cameras monitoring the parking area. In our study, all these cases are tackled.

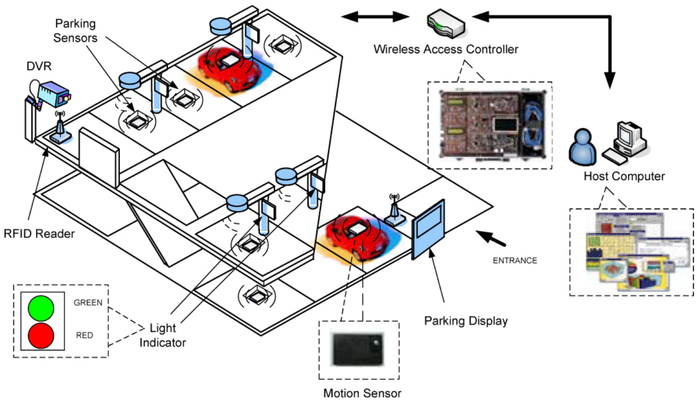
In this paper, we provide a collision verification method to extend our previous works in the ubiquitous parking management system (UPMS). Our previous studies were car presence detection [1] and location technique [2] using wireless sensors in parking management field. This paper studies the collision detection which uses a motion sensor to detect the collision and provide comprehensive information, and a video tracking is used to perform the oriented object box (OBB) intersection test for verification. The collision detection uses several sensors to perform the location technique for accurate details of collision information and verification provides a fast OBB intersection test by using the nearest edge point of each car to separate the cars.

2. Real-Time Collision Detection and Video Tracking

Collision detection is fundamental in video games, virtual-reality modeling, geometric modeling, and robotics to determine collisions between objects. In [3], the collision detection is classified into two types which are the computational geometry and image processing algorithms. To translate the intersection of objects in the graph, bounding volume is needed to encapsulate one or more objects [4]. The efficient algorithm requires a simple shape to have a cheaper test in determining the intersections. The Axis-Aligned Bounding Boxes (AABBs) is the fastest in computing bounding volumes but is not accurate to determine the intersections. Another test is the Oriented Bounded Boxes (OBB) which is more accurate compared to AABBs and less complex than other tests. An exact test for OBB-OBB intersection can be implemented in terms of the separating axis test. Two OBBs, represented by A and B, and radii of each object srepresented by r, are separated if,

$$|T \cdot L| > r_A + r_B \tag{1}$$

where L is the translated axis and T is the distance between the centers of the two objects. For OBBs it is possible to show that at most 15 of these separating axes must be tested to correctly determine the OBB overlap status. It is also possible 6 of the test are enough to determine the separation. In [4], an AABB collision test is used for humanoid robot which checks more than 100 collision pairs in



(Fig. 1) UPMS in a two level parking establishment.

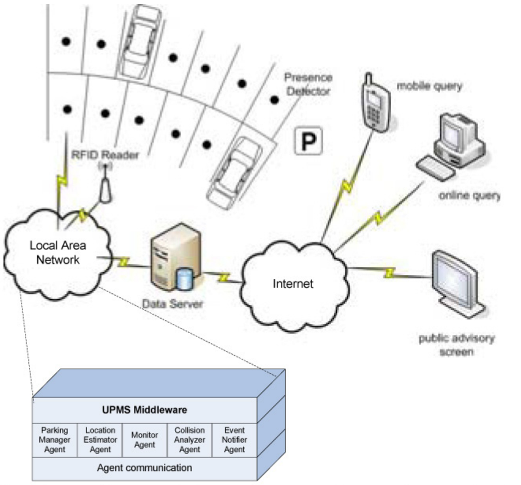
real-time. In [5], a camera is used to estimate the terrain and calculate the possible collision. A corresponding response is included after detecting the collision. Real-time detection is also popular in preventing collision in roads [6]. In the collision event, a sensor that senses the change of position of a non-moving car is needed. However, in identifying the cause of collision, we need to determine the details like position, speed and direction of cars.

The object or video tracking estimates the positions and other relevant information of moving objects in the image sequences. The model selected to represent object shape limits the type of motion or deformation it can undergo. We summarized the categories of object tracking in [7] into three: (a) points, (b) kernel and (c) silhouette. In the point tracking, objects detected in consecutive frames are represented by points, and the association of the points is based on the previous object state which can include object position and motion. The Kernel tracking refers to the object shape and appearance, e.g. a rectangular template or an elliptical shape with an associated histogram. The motion of objects is usually modeled by a parametric transformation such as translation, rotation, and affine. The silhouette tracking is performed by estimating the object region in each frame. Given the object models, silhouettes are tracked by either shape matching or contour evolution.

3. Ubiquitous Parking Management System based on Multi-Agent Communication

The UPMS [6] is consisted of three layers: ubiquitous network, middleware and application services layers. In the ubiquitous network, represents the physical networks of different sensor devices and computers communicating in the wireless environment. The components in the middleware layer are transparently executing for the efficiency on managing data from the ubiquitous network layer. Interaction of clients and application services are also handled by the middleware layer. The application service layer is consisted of services for parking management system.

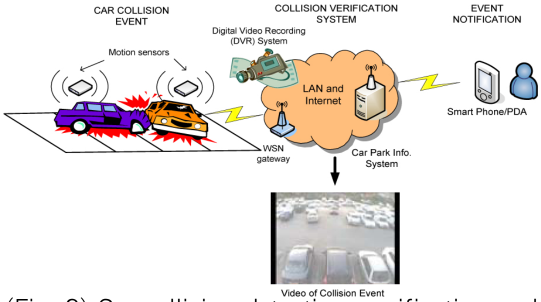
Figure 1 is a design of UPMS emphasizing a two-level parking area using the following devices: car presence sensor, RFID readers, parking display, wireless access controller, digital video recorder and motion sensor. Car owners will be provided with a motion sensor before entering the parking area. Before a car enters the parking area, a display of available slot is projected in the screen of the parking display



(Fig. 2) Interaction of the component devices in parking events based on the UPMS middleware.

monitor. On entering the parking area, the RFID readers scan the identification of car. The drivers will be guided by the light indicators on top of parking slots which indicates vacant (green) or not (red). After choosing an empty slot, a car that has parked is detected by a car presence sensor. The car presence sensor sends a message to the wireless access controller to record the parking event and to store the data which includes the time of parking, car presence sensor ID and identification of the car. A car owner should turn-on the motion sensor after parking and leaving the car, and then turn-off before exiting the parking area. Also, a digital video recorder records the video of parking events inside the parking area which is also used in collision verification method. To identify a car, a camera is installed in the gate to get the image of the plate number and this will be associated to the motion sensor and object tracking technique.

In Figure 2, each presence sensor/detector transmits message to process in the server to log the parking event using the wireless network. On the other hand,



(Fig. 3) Car collision detection, verification and notification methods

smart phone and display devices use the Internet to request information. Agents in the UPMS middleware are shown at the bottom of Figure 2. The UPMS middleware uses multi-agent approach to provide an intelligent distribution of task and efficient information dissemination within the system. The multi-agent components are defined as follows:

- **Parking Manager Agent** – stores the information gathered from all agents and processes the request of information of an agent.
- **Location Estimator Agent** – used in location technique for collision detections of cars inside the parking area.
- **Collision Analyzer Agent** – detects the collision inside the parking area. The result from detection is sent to the PMA to verify the collision.
- **Monitor Agent** – displays the current available slots and other information for the car that entering the parking area.
- **Event Notifier Agent** – informs the owner of the important events inside the parking area.

4. Collision Detection and Verification System

The collision analyzer agent determines the collision event and then the parking manager agent informs the Parking Information System (PIS). There are three procedures of the proposed collision verification method which are: 1) *collision sensing*, 2) *collision verification* and 3) *car owner notification*. In Figure 3, the interaction of the components in UPMS for the car collision event is shown.

4.1 Collision Event Sensing

A motion sensor is used to detect the collision event and each car inside the parking area is provided with a motion sensor. A *parking* status is activated manually by a car owner every time his or her car is parked. A button is used to activate and a light indicates the status. The motion sensor is in *parking* status until the car owner returns to the car and manually changes the status to *leaving*. In the case of a *non-moving car* (v_1), a collision is detected by comparing a threshold value (Φ) from the output differences of the 3 axis values (x, y, z) over time (t) of the motion sensor represented by $K=\{x,y,z\}$ and each axis has $x=\{x_1,\cdots,x_t\}$, $y=\{y_1,\cdots,y_t\}$ and $z=\{z_1,\cdots,z_t\}$.

$$X=\sum_{t\in K}^{K} x_t - x_{t-1},\ Y=\sum_{t\in K}^{K} y_t - y_{t-1},\ Z=\sum_{t\in K}^{K} z_t - z_{t-1} \tag{2}$$

$$v_i = X+Y+Z \tag{3}$$

Equation 3 shows the total value in Equation 2 which represents the current motion axis value differences in sensor *I*. If the sensor senses a movement and $v_i > \Phi$, the motion sensor sends a message to the wireless controller in the parking area to process the collision verification method. In the case of a *moving car* (v_1) that collides to a non-moving car, the information of direction is acquired by the motion sensor using the *x* and *y* axis. The motion sensor is still gathering information in the *leaving* state. The motion sensor on a non-moving car will process the comparison of the motion value differences and the threshold. This is also the case that the motion sensor of a non-moving car is activated while in the moving car, the sensor is in *leaving* status. The motion sensor is still processing the x, y and z axis values in a moving car and this information is used to detect the position of a moving car using the direction readings of coordinate x and y which is listed in Table 1.

〈Table 1〉 Rules for the collision information

Rules	**x**	**y**	**Direction**
1	0	POS	*FORWARD*
2	POS	POS	*FORWARD RIGHT*
3	POS	0	*RIGHT*
4	POS	NEG	*RIGHT BACKWARD*
5	0	NEG	*BACKWARD*
6	NEG	NEG	*LEFT BACKWARD*
7	NEG	0	*LEFT*
8	NEG	POS	*LEFT FORWARD*

In Table 1, the *POS* is a positive value and *NEG* is negative value of the motion sensor. A zero (0) represents a constant movement on a specified axis and when both x and y are 0 then the car is not moving. Rules associated with the coordinate values (x, y) are defined in Table 1 which are used for the direction information of a moving car. E.g., if the values of a current moving car are x(-5) and y(-3), means that it is moving left backward. The values from Table 1 represent the direction of a moving car and this information is sent after it detects the collision event. Also, this is used as comprehensive position details of cars in the collision event. Using this information, the verification method is processed and a location technique is used to verify the position of the cars. The location method using the CARFID

(Table 2) Information from *non-movingcar*(v_1)

Time (Sec)	Position (CARFID)	Direction (Motion Sensor)
1	x=200, y=200	***NO MOVEMENT*** (x=0,y=0)
2	x=200, y=200	***NO MOVEMENT*** (x=0,y=0)
3	x=195, y=205	***FORWARD LEFT*** (x=-5,y=5)
4	x=193, y=206	***FORWARD LEFT*** (x=-2,y=2)
5	x=193, y=206	***NO MOVEMENT*** (x=0,y=0)

(Table 3) Information from *movingcar*(v_2)

Time (Sec)	Position (CARFID)	Direction (Motion Sensor)
1	x=265, y=125	***NO MOVEMENT*** (x=0,y=0)
2	x=240, y=160	***FORWARD LEFT*** (x=-25,y=0)
3	x=220, y=170	***FORWARD LEFT*** (x=-20,y=10)
4	x=218, y=172	***FORWARD LEFT*** (x=-2,y=2)
5	x=218, y=172	***NO MOVEMENT*** (x=0,y=0)

[2] starts calculating the distance of each car presented in Equation 4, where P_{tx} is the transmission power, a is the path loss model, P_{rcvd} is the signal strength is used to solve the rordistance.

$$P_{rcd} = c\frac{P_{tx}}{r_i^{\alpha}} \Leftrightarrow r_i = \alpha\sqrt{\frac{cP_{tx}}{P_{rcd}}} \qquad (4)$$

The distance value from Equation 4 is used for location estimation procedure in calculating the multi-lateration [8] of the location estimation of each car. The *r* is the hypotenuse of a triangle between two known coordinates (*x* and *y*) of an anchor or node (*n*) and Equation 5 is a matrix to calculate the multi-lateration method.

$$2\begin{bmatrix} x_n - x_1 & y_n - y_1 \\ \vdots & \vdots \\ x_n - x_{n-1} & y_n - y_{n-1} \end{bmatrix}\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} (r_1^2 - r_n^2) - (x_1^2 - x_n^2) - (y_1^2 - y_n^2) \\ \vdots \\ (r_{n-1}^2 - r_n^2) - (x_{n-1}^2 - x_n^2) - (y_{n-1}^2 - y_n^2) \end{bmatrix} \qquad (5)$$

However, the output from the received signal *r* can have errors because of the obstructions and interference parking area. To minimize the errors in getting the approximate value from the procedure, the system needs more information by setting additional sensor nodes. These sensors are represented by *n* in the Equation 5. These are processed in the location estimator agent and after knowing the locations, near cars from the collision are processed in the verification. The position details of each selected car are determined by basing from its locations over time which will be included and stored in U = {v_1, v_2} and sent as information for collision. Table 2 and Table 3 show the collision event information using the details from the location technique and motion sensor. The shaded row signifies the time of collision event and the information two seconds before and after the collision event is shown.

4.2. Collision Verification

In this procedure, it assumes that the system already received the information from the motion sensor indicating that there was a car collision. The proposed method will process the information from the motion sensor and queries the video frames of the collision event to identify the cars involved in the collision which is important for recording and informing car owners about the event. We define *t* for the time of collision event and *pt* for the specified period of time to query the video. The system queries from initial time t_i where $t_i = t - pt/2$ to the end time t_e where $t_e = t + pt/2$. All images from t_i to t_e is used in the video tracking technique. At first, moving objects are identified by the intensity of images which simply uses a threshold value to analyze the intensity in each pixel. After separating the moving objects in the image, the method merges object pixels that are close to each other to create blobs. The OBB that highlights a moving car is also processed. The blob analysis groups the pixels that represent a portion of a car and then it determines the bounding boxes of the blobs of pixels. A car image represented by an OBB is described by a vector *z* including center coordinates (*cx*, *cy*), orientation angel (Θ) and size (*dx*, *dy*) in Equation 6.

$$z_A = [cx, cy, \theta, dx, dy] \qquad (6)$$

The contour lines of OBB are refined by the blob analysis and a method based on rotating calipers aligns the object orientation. In the final step, it removes insignificant bounding boxes (too small or too large) so that the OBBs only represent cars. The Kalman Filter [9] is used to determine the accurate location of each car in every frames of a video. To determine the locations of specific car from one frame to another, the proposed method compares the predicted location of the bounding boxes with the detected locations. The OBB information is stored in Z = {z_A, z_B} to be used in verification funtion. The bounding box encapsulates the moving objects to process the OBB intersection test of objects (cars) in the graph. Two OBBs, represented by *A* and *B*, and radii of each object represented by *r* are compared using Equation 1. The current positions of cars are parallel with the line drawn from the previous and current coordinate of the bounding box of a car. At the start of the separation test, the nearest edges from other moving cars or objects are determined to draw the separation line shown in Equation 7. The Euclidean distance of the edge point *j* of object *i* and the object center *k* is calculated. After choosing the edges from Equation 7, these edges are used to calculate the separating points. The *r* length is calculated by the scalene triangles formed by the length from the centers of objects ($edge_{eA}$ and $edge_{eB}$) in Equation 8 and the translated axis *L'* is calculated by the distance over the angle given from z shown in Equation 9.

$$e_A = \min\{\sum_{j=1}^{A} |e_{ij} - c_k|\} \qquad (7)$$

$$r_A = \frac{2d(c_A, e_A)}{d(e_A, e_{A'})} \qquad (8)$$

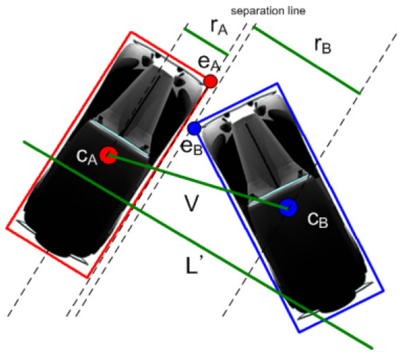
$$L' = \frac{V}{\cos\theta} \qquad (9)$$

$$|V \cdot L'| > r_A + r_B \qquad (10)$$

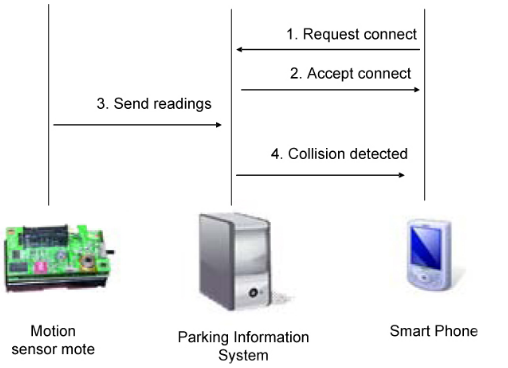
The proportion value from adding the two lengths in each edge length is used to calculate the r_A. The proportion value of $edge_{eA}$ is multiplied to the *L'*. Also, the same procedure is done to other object. This procedure is done twice from object *A* to *B* and *B* to *A*. If the result of two tests is either true from Equation 10 then it is said there are no collision happened.

Lastly, the verification compares with the information *V* gathered from the motion sensor to the information of the object tracking *Z*. We calculate both values in the detection and verification by $e=Z(V)=\{e(z_A,v_1)+e(z_B,v_1)\}$ is the function of comparing both information in the collision verification process where *e* is error value. This method can analyze the situation of collision as:

- A moving car collides to another non-moving car if one of the cars is determined as moving and other car is at a parking status.



(a)



(b)

(Fig. 4) Separation lines of two cars using the proposed nearest edges (a) and interaction of sensors to inform the car owner through the PIS (b).

- A moving car collides to another moving car if both detected collision and both cars are determined as moving.
- An object collides to another moving car verified in the object tracking.

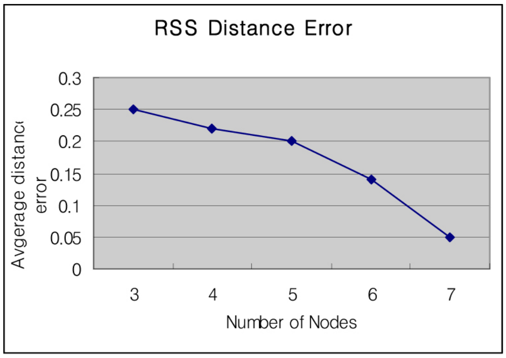
4.3. Collision Notification by the Parking Information System

Figure 4 illustrates the separation procedure between two cars using their current location which is calculated in the object tracking of collision verification. These calculations are done by the collision analyzer agent. The collision analyzer agent

(a) (b)

(Fig. 5) CARFID with motion sensor (a) and PIS interface for mobile clients (b).

communicates to the parking manager agent about the event and sends the information of the collision event which includes the cars affected by the collision and the time it occurred. The Parking Information System (PIS) is a subcomponent of the UPMS which manages the dissemination of vital information to the car owners and management. The PIS is informed about the collision event occurred inside the parking area using the proposed collision detection and verification. After the collision analyzer agent identified the cars that collided, the parking manager agent processes the verification and reports the car collision event to the PIS. In Figure 5, the interaction of the devices is shown. (1) First, the smart phone of the car owner connects to the PIS and then (2) the connection will be accepted after the user verifications. The PIS continues on monitoring of the collision event. If the sensor detects a collision then it sends the reading to the gateway sensor (3) and process the collision verification. After the verification and informing PIS, it sends the notifications to the respective car owners which are involved in the collision (4).



(Fig. 6) Distance error in meters using the location estimation based on RSS by increasing the number of nodes.

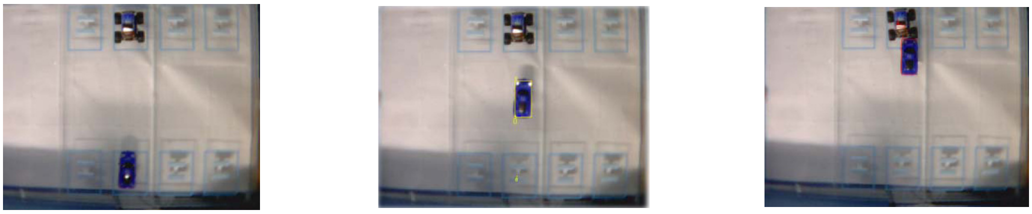
5. Implementations

The implementation of UPMS used wireless sensor motes which is a 2.4 GHz IEEE 802.15.4 compliant RF transceiver, car parking platform and a web camera. The proposed collision verification method and the PIS were integrated in the UPMS modules. The program for the motion sensor and CARFID were coded and designed using nesC and the agents were programmed in Java. The area of the CARFID is 8.5 cm by 5.5 cm which is equal to the size of an ATM card and the thickness is about 1 cm shown in Figure 5a.

Figure 5a shows the CARFID with the motion sensor. The CARFID was used for the collision sensing and identification of cars [2]. If collision event is detected then it sends the information to UPMS to process the proposed collision verification and notification. The parking information system was designed using Java shown in Figure 5b. The interface for clients is consisted of commands to locate a car, receive event messages, clearing the current message and resetting the connection to the PIS server. The interface is connected to PIS server to receive notification.

The accuracy of the location technique in determining the position of cars is critical in the procedure of the collision detection. This includes the information needed to determine the current positions of the car by the previous and current coordinates, and the calculation of the separating lines between cars. An area of 10 meters by 10 meters is used to simulate and evaluate the location scheme. Also an obstruction or interference was introduced in adding the fifth sensor. In Figure 6, the distance error of the multi-lateration by gradually adding sensors in processing the proposed scheme is determined by a line graph.

In Figure 7, the images of video in verifying the collision event are shown. The yellow boxes represent moving car while the red boxes represents colliding cars. The procedures of object tracking and calculation of OBB intersection are processed in the proposed collision verification method. The OBB intersection is performed only 4 tests using the nearest edge of each car.



(a) (b) (c)

(Fig. 7) Process of object tracking technique in video: Non-moving cars (a) a moving car (b) collision of two cars (c).

6. Conclusion and Future Works

This paper showed the design of ubiquitous parking management system (UPMS) where wireless sensors were used to detect important events in the parking area. Also, the design was based on multi-agent system to disseminate efficiently the information through the system. A collision verification method was proposed to detect and to analyze the collision event, and notifies car owners. The detection method uses the information in motion sensors for comprehensive details of position and directions of a moving car and the verification processes an object tracking technique with a fast OBB intersection test. After the verification, the information is sent to concerned car owners. The collision verification provided a fast OBB intersection test by using the nearest edge point of each object to separate the objects.

This paper contributed on providing comprehensive information for the collision event and the minimization of calculating the separation lines of collision detection by introducing the nearest edges point for the collision verification. The optimization of determining the position of cars and location estimation are still considered as future work. Managing the problem of collision detection within the error range limit will be tackled in the future work.

References

[1] 마테오 로미오, 이재완: 유비쿼터스 센서네트워크를 위한 퍼지시스템 기반 적응형 센싱, 인터넷정보학회논문지, 제9권 제3호 (2008) pp. 51-58.

[2] Mateo, R.M., Yoon, I., Lee, J.: Location Technique based on Pattern Recognition of Radio Signal Strength using Active RFID, Proc. of IWRT(2008) pp.7-18

[3] Ericson, C.: Real-Time Collision Detection, Morgan Kaufmann Publishers (2005) pp. 101-107.

[4] Ullrich, T., Funfzig, C., and Fellner, D.W.: Two Different Views on Collision Detection, IEEE Potentials, Vol. 26, No. 1 (2007) pp. 26-30.

[5] Okada, K., Inaba, M., and Inoue, H.: Real-time and Precise Self Collision Detection System for Humanoid Robots, Proc.of IEEE International Conference on Robotics and Automation (2005) pp. 1060-1065.

[6] Luo, F., Zhong, E., Huang, Y.F., Guo, H., and Cheng, J.L.: Real-time Collision Detection and Response in Virtual Global Terrain Environments, Proc. IEEE Computer-Aided Industrial Design & Conceptual Design (2009) pp. 2257-2262

[7] Veenman, C. Reinders, M., and Backer, E.: Resolving Motion Correspondence for Densely Moving Points, IEEE Trans. Patt. Analy. Mach. Intell. Vol. 23, No. 1 (2001) pp. 54-72.

[8] Karl, H. and Willig, A.: Protocols and Architectures for Wireless Sensor Networks. John Wiley & Sons, Ltd. (2005) pp. 238-239

[9] Jang, D.S., Kim, G.Y., Choi, H.I.: Model-based Tracking of Moving Object, Pattern Recognition, Vol. 30, No. 6, (1997) pp. 999-1008.

◐ 저 자 소 개 ◑

Romeo Mark Mateo

2004년 West Visayas State University, Philippines BS in Information Technology
2007년 Kunsan National University, South Korea, Master of Engineering major in Information and Telecommunications
2007년~현재 Kunsan National University, SouthKorea, Graduate student in Ph.D course
관심분야 : Distributed systems, data mining, fuzzy systems, multi-agents, ubiquitous sensor networks, cloud computing
E-mail : rmmateo@kunsan.ac.kr

양 현 호 (Hyunho Yang)

1986년 광운대학교 전자공학과 졸업(학사)
1990년 광운대학교 대학원 전자공학과 졸업(석사)
2003년 광주과학기술원 정보통신공학과 졸업(박사)
1989년~1990년 삼성SDS 근무
1991년~1997년 포스데이타(주) 근무
1997년~2005년 순천청암대학 근무
2005년~현재 군산대학교 정보통신공학과 교수
관심분야 : 무선데이터통신, RFID/USN etc.
E-mail : hhyang@kunsan..ac.kr

이 재 완 (Jaewan Lee)

1984년 중앙대학교 이학사-전자계산학
1987년 중앙대학교 이학석사-전자계산학
1992년 중앙대학교 공학박사-전자계산학
1996년 3월~1998년 1월 한국학술진흥재단 전문위원
1992년~현재 군산대학교 교수
관심분야 : 분산 시스템, 운영체제, 실시간 시스템, 컴퓨터 네트워크, 클라우드 컴퓨팅 등
E-mail: jwlee@kunsan.ac.kr