

# **The Autonomous Shopping-Guide Robot in Cashier-Less Convenience Stores**

Po-Han Lin<sup>1,\*</sup>, Cheng-Yi Lin<sup>1</sup>, Chen-Ting Hung<sup>1</sup>, Jen-Jee Chen<sup>1</sup>, Jia-Ming Liang<sup>2</sup>

<sup>1</sup>Department of Electrical Engineering, National University of Tainan, Tainan, Taiwan

<sup>2</sup>Department of Computer Science and Information Engineering, Chang Gung University, Taoyuan, Taiwan

Received 19 March 2019; received in revised form 13 May 2019; accepted 08 July 2019

DOI: <https://doi.org/10.46604/peti.2020.3961>

## **Abstract**

In recent years, several cashier-less convenience stores have appeared, including Amazon's. A store without cashiers will become a trend in the near future. To substitute the employee in traditional stores, this research proposes and designs an autonomous shopping cart robot to guide customers' purchase according to the requested shopping list to enhance their shopping experience. The core techniques of the system are the autonomous driving robot, the traffic control center for robots and the dynamic route planning algorithm. The robot is a self-propelled vehicle developed by ROS (Robot Operating System), in which we achieve the automatic driving via image recognition, April tag identification and driving direction guidance from the path planning and traffic control services. This enables the robot to lead customers to find their commodities following the preplanned route. In conjunction with the vocal service, the robot can notify the customer when arriving at each commodity, he or she plans to buy. We also design a light APP for customers to easily set up and manage their shopping list, call for the robotic shopping cart's help, and interact with the shopping cart robot. To enhance the shopping experience of customers, we design the dynamic route planning genetic algorithm to dynamically plan the shopping route according to the customer's request and the traffic condition. Experiments show that our genetic algorithm can provide the most stable performance and always get efficient shopping route planning in a limited time compared to other methods.

**Keywords:** ROS (Robot Operating System), wireless networks, navigation, autonomous driving, cashier-less convenience store, route planning, genetic algorithm

## **1. Introduction**

The development of artificial intelligence and automation technologies has been advancing by leaps and bounds over the recent years. This also helps several emerging applications make big progress, including the autonomous driving vehicle. To reduce the setup cost for researchers and in school students and help them to test their designed or learned autonomous driving algorithms in a quick manner, Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory (CSAIL) designs an open source platform, called Duckietown [1]. This helps more researchers and students to be involved in the field of autonomous vehicle technology [2] and has been confirmed from the actual case of education [3]. In this paper, we aim to improve the quality of shopping in cashier-less convenience stores with autonomous vehicle-based robots. This expands the possibilities of autonomous vehicle technology to the indoor application, not only the common outdoor car application.

In [4], a map-based navigation method with indoor vision is proposed. To improve the adaptability of the robot, a small vehicle is equipped with a camera to capture the visual information of the environment which helps the robot to sense the

---

\* Corresponding author. E-mail address: [cliffabc2004@gmail.com](mailto:cliffabc2004@gmail.com)

environment changes so that the vehicle could smoothly navigate along the planned route. On the other hand, autonomous driving is an emerging and hot research area [5]. It is a complex system, so the functional modules are highly dependent. The stability of the system is very important. There are many researchers using ROS for the implementation of automatic driving and prove that this is feasible [6]. The authors in [6] demonstrate that ROS is suitable for developing autonomous driving applications in various configurations from the perspectives of ROS characteristics, communication overhead, and advantages and disadvantages comparisons.

Amazon's Cashier-Less Convenience Store [7-8] has been debated since last year. A store without any cashier will become a trend in the near future. To guide customers' purchase in the cashier-less convenience store and enhance their shopping experience, we propose to design an automatic shopping cart robot to substitute for the employee in traditional stores. To realize such an auxiliary robot helping customers' shopping in stores, a novel framework which integrates the autonomous driving system for navigation, dynamic route planning genetic algorithm and APP-based human-machine interaction interface is presented. For dynamic route planning, we set up a cloud server to calculate the shortest route according to the customer's requested shopping list by using a genetic-based algorithm. In this work, our contributions are threefold:

- (1) Considering the trend of more cashier-less stores being built, our system provides a brand-new way to enhance customers' shopping quality and experience. The proposed framework is a general solution such that it can work in most cases.
- (2) A prototyping system is completed and developed. In the prototype, the hardware setup is based on the low-cost Raspberry Pi2 while the software is totally open source. Even the prototyping cart is made by using 3D printing. The setup cost and the maintenance fee are acceptable for most organizations and groups. New functions, algorithms, mechanisms, and structures can be enhanced or added on it without difficulties, which guarantees the scalability of our system.
- (3) To realize the autonomous driving and navigation functions over the robotic shopping cart, we implement image recognition capability at the local cart and develop dynamic route planning genetic algorithm at a central server. The former allows the robotic cart to respond to road conditions and emergencies immediately; while the latter enables the best route navigation to be provided to the customer by considering the global situations of the store. Experiments show that the system works correctly and simulation results demonstrate that our proposed algorithm is stable and efficient in time complexity, route efficiency, and performance stability.

## 2. System Architecture and Proposed Methods

In this Section, we will first introduce the system architecture of the whole system. Then, the operation procedure of the ROS-based robotic shopping cart will be described. Finally, the dynamic route planning genetic algorithm will be illustrated.

### 2.1. System architecture

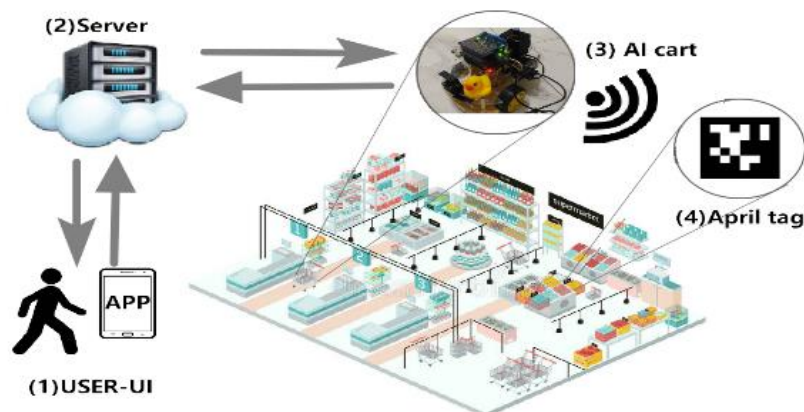


Fig. 1 System illustration

In the proposed system, as shown in Fig. 1, customers have to connect with the shopping cart by scanning the QR code on the cart with our developed APP. After connecting to the cart robot, the customer can browse the commodities via the APP and create his or her shopping list. Upon receiving the shopping list of the customer, the central server will plan the most efficient route for the customer to get all his/her checked commodities. Then, the shopping cart robot will start to navigate its customers. Each kind of commodity in the store is tagged with an AprilTag. Once the cart robot identifies an AprilTag that belongs to one of the unordered commodities of the customer, it will stop in front of the commodities and notify the customer with voice. When the user is ready for the next commodity, the robotic shopping cart will continue its navigation. Moreover, our APP allows the customer to interact with the cart robot or issue a command to the robot. Furthermore, the system administrator can see what the robot sees via the remote monitoring subsystem to check the status of each robot. The remote monitoring subsystem can also provide the information of robot distribution in the store for the administrator. Fig. 2 shows the system architecture, which includes the APP, central cloud server, shopping cart robot, and the remote monitoring subsystem. In next subsection, we will describe the system operation procedure in more detail.

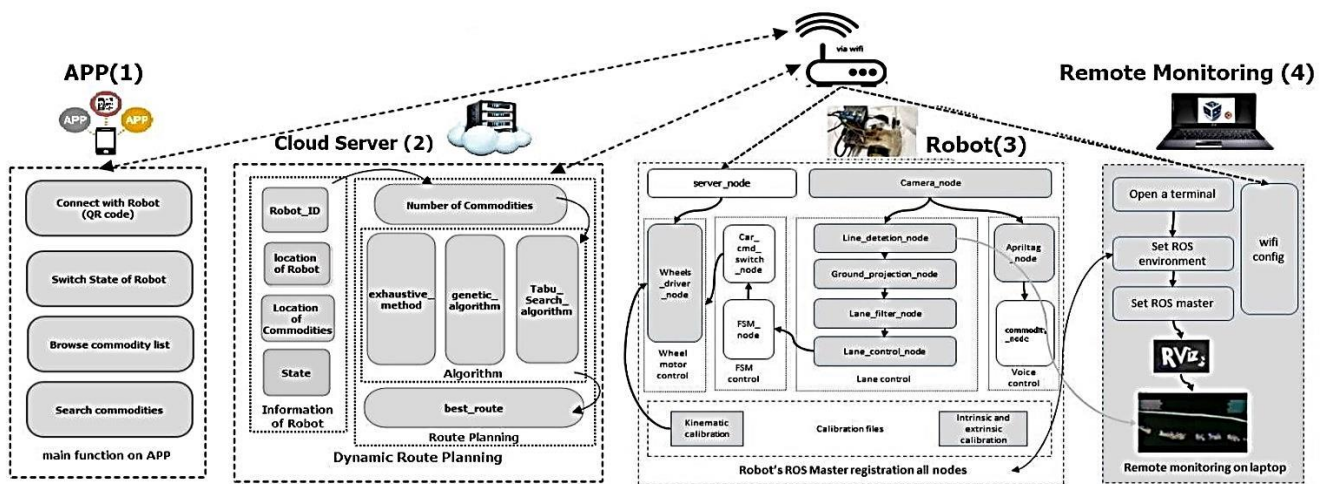


Fig. 2 System architecture

## 2.2. Operation procedure

The robot is equipped with a fisheye camera, which keeps recording the road image and then passes the images to the Lane Control Module and Voice Control Module. As shown in Fig. 2(3), the Lane Control Module, FSM Control Module, and Wheel Moter Control Module are used to realize the lane-following function of the robot. On getting the compressed image from the camera node, the Line detector node uses the OpenCV library to identify the desired color on the road and then passes the image to the Ground projection node. The Ground projection node marks and converts the image to compute exact coordinates of the image in the space. Afterwards, the information enters the Lane filter node and then calculates the driving direction according to the identified lane and coordinates. The driving direction is sent to the Lane control node and the node will convert the direction into the dedicated car\_cmd. The operation procedure then enters the FSM Control Module. Upon the receipt of the car\_cmd from the Lane Control Module, the FSM node switches or maintains the state accordingly. Inside the module, the FSM and Car cmd switch nodes communicate with each other to keep the FSM status in the Car cmd switch node up to date. Specifically, the Car cmd switch node is responsible to drive the Wheel driver node in the Wheel Motor Control Module to manipulate the motors according to the state of FSM. On the other hand, upon the receipt of the recording image from the camera node, the AprilTags node in the Voice Control Module processes the image and recognizes whether there is any AprilTag in the image. If yes, the ID of the tag will be conducted and then be forwarded to the commodity\_node. On receiving the tag information from the AprilTag node, the commodity\_node will check whether the tag is for a commodity or an intersection. If it is a tag of a commodity and the commodity is in the customer's shopping list, the robotic cart will notify the customer and wait for the command of the customer. On the contrary, if the tag means an intersection, the robot will send

the content of the tag to the central server and waits for the instruction from the server. Possible instructions include turning right, turning left, going straight and making a U-turn.

### 2.3. Dynamic route planning

To let the customer be able to find all his or her requested commodities in short time. We design a dynamic route planning genetic algorithm to plan the shortest route in the cashier-less convenience store for the user according to his or her selected commodities in APP. The route planning problem can be modeled by using a graph  $G = \{V, A\}$ , where  $V = \{v_1, \dots, v_n\}$  is the set of vertices, each  $v_i \in V$  is either an intersection or a commodity, and  $A = \{(v_i, v_j): v_i, v_j \in V, i \neq j\}$ , each arc  $(v_i, v_j) \in A$  has a weight (or length)  $c_{ij} \geq 0$ . The problem is similar to but not completely equal to the Travelling Salesman Problem (TSP), which is a well-known NP-hard problem [9]. But we still can refer to those previous methods which solve the TSP problems. Since the response time is important to the customers, our designed algorithm has to be completed in limited time and the planned route can be sub-optimal but close to the optimal solution. To achieve this, we have designed a dynamic route planning scheme based on genetic algorithm.

### 2.4. Dynamic route planning genetic algorithm

The genetic algorithm models a process of evolution of a population of individuals. Each individual  $G_k$  is a feasible solution to the route planning problem. The components of an individual are called genes. Here, in this work, each gene  $g_k(l)$ ,  $l = 1..M$ , is one of the  $M$  selected commodities of the customer. For any two neighbor genes,  $g_k(l)$  and  $g_k(l+1)$ , we adopt the shortest length route, such that the cost  $c_{g_k(l), g_k(l+1)}$  of moving from commodity  $g_k(l)$  to  $g_k(l+1)$  is the shortest route length from  $g_k(l)$  to  $g_k(l+1)$ . So, the total cost of individual  $G_k$  is

$$C_{tot}(G_k) = \sum_{l=0}^{M+1} C_{g_k(l), g_k(l+1)} \quad (1)$$

where  $g_k(0)$  and  $g_k(M+1)$  are the starting point (i.e., entrance of the store) and ending point (i.e., checkout counter), respectively. The steps of our dynamic route planning genetic scheme are described as below.

- (1) Create initial populations  $G_k$ ,  $k = 1..K$ . Set  $t = 1$ .
- (2) Repeat steps 2.1-2.5 until  $t = T$ :
  - (2-1) Remove the individuals with  $C_{tot} > CTH$ .
  - (2-2) Choose two parent individuals  $G_x$  and  $G_y$  from the remaining populations. Note that for each  $G_k$ , the chosen probability is proportional to  $1/C_{tot}(G_k)$ .
  - (2-3) Apply crossover to  $G_x$  and  $G_y$  and then obtain new individuals  $G'_x$  and  $G'_y$ . Repeat steps 2.2 and 2.3 until the total amount of parent individuals and offspring individuals being equal to or greater than  $K$ .
  - (2-4) For each individual  $G_k$ , each of its gene  $g_k(l)$ ,  $l = 1..M$ , has a probability of  $p_m$  to apply mutation.
  - (2-5) Set  $t = t + 1$ .
- (3) Choose the individual with the least route length among the remaining populations as the solution.

## 3. Prototype Implementation and Experimental Results

In our designed cashier-less convenience store environment, user can browse the commodities by APP and request the DuckieBot to find the selected commodities. DuckieBots follow the central server planned route to guide users in the store. Once the DuckieBot finds an AprilTag of commodities, it stops and uses the voice function to notify the customer. The user can use APP (as shown in Fig. 3) to control the state of shopping cart.

We use Duckiebot, a device that serves as the primary client for autonomous driving in Duckietown [4], as the basic architecture for navigation robots. Our main hardware and software are based on the Raspberry Pi 2 B+ Ubuntu operating system. Other hardware includes a Adafruit DC & Stepper Motor HAT, Adafruit 16-Channel PWM/Servo HAT, fisheye camera, car module, mobile power and the internal system of the robot is open-source ROS (as shown in Fig. 4). A demo video is uploaded and the link is as below: <https://ez2o.co/67hH5>.

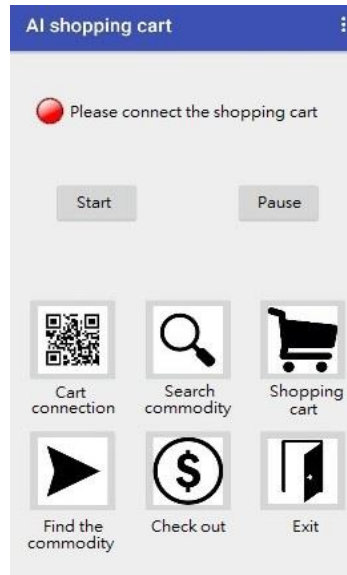
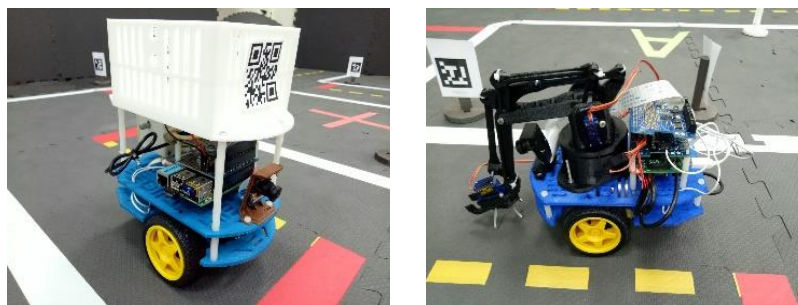


Fig. 3 Our developed App



(a) Robotic shopping cart model (b) Shopping cart with machine hand

Fig. 4 The hardware prototyping of our robotic shipping carts

In the experiments, we compare our route planning Genetic Algorithm (GA) with other methods. The methods include Exhaustive Search Algorithm (ESA), Simulated Annealing Algorithm (SAA), Greedy method and Tabu Search method (TS).

As shown in Fig. 5 and Fig. 6, ESA always gets the shortest route but spends the longest execution time. The execution time increases exponentially as the number of selected commodities increases. The lengths of the derived routes in GA, SAA, and TS are close to ESA while the execution time of them are much less than ESA.

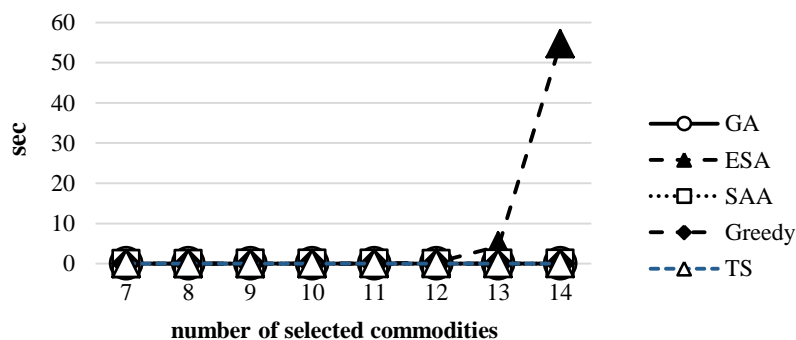


Fig. 5 The effect of the number of selected commodities on the execution time

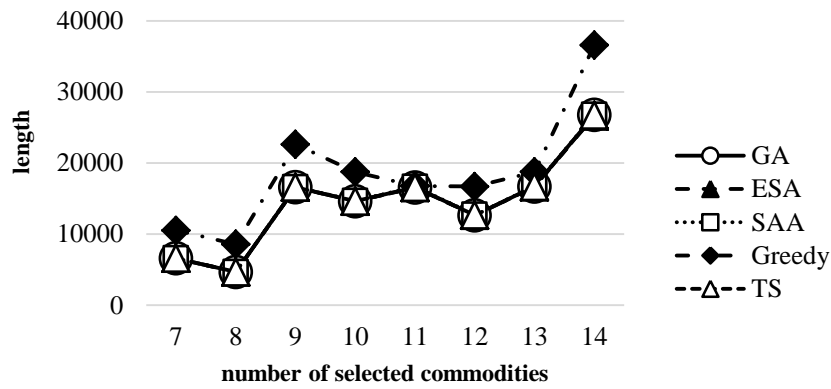


Fig. 6 The effect of the number of selected commodities on the route length

As shown in Fig. 7, we can see GA always get the shortest route length compared to other methods. Compared to GA, the performance of other methods is unstable. SAA and TS can get almost the same performance when the number of selected commodities is less than 20, but the outcome becomes unstable when the number of selected commodities are more than 20. As shown in Fig. 8, the more the generations we use in GA, the better the solutions to route planning we get. We can see that the performance converges when the running generations are more than 200. The improvement of route planning results are very limited, even if we spend more time on calculation. So, we choose to set the number of generations to 200 in GA.

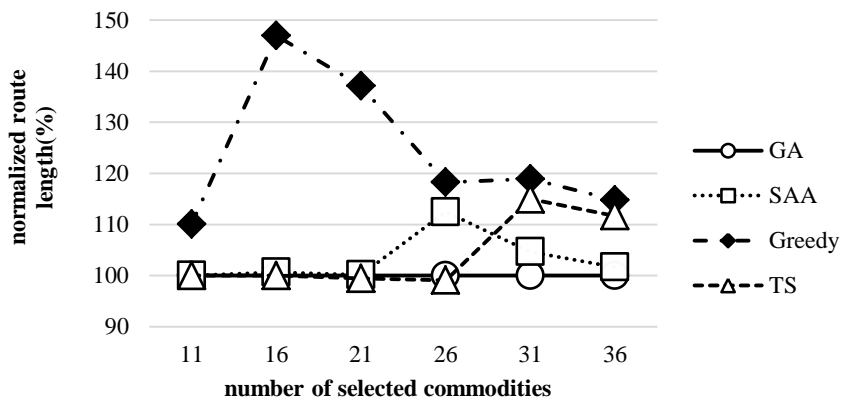


Fig. 7 The effect of the number of selected commodities on the normalized route length (the normalized route length of each method = (route length of each method/route length of GA) \* 100%)

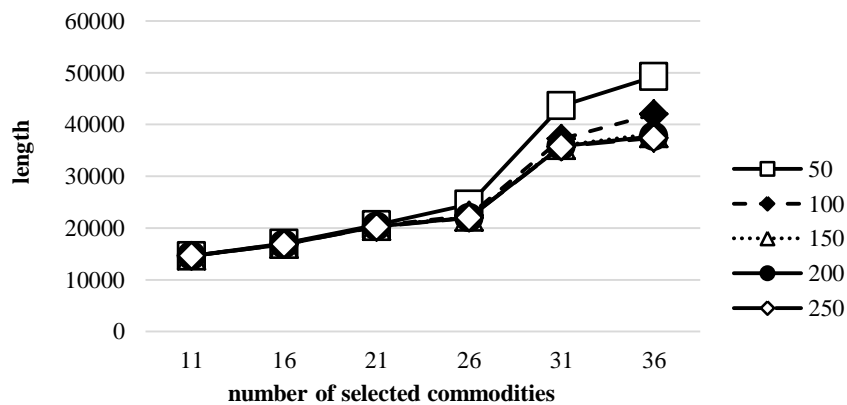


Fig. 8 The effect of running generations of GA on the route length

### 4. Conclusions

In this paper, to improve the shopping quality of customers in cashier-less convenience stores, we design an automatic robotic shopping cart which integrates image recognition, dynamic route planning, and APP as the human-machine interface.

The shopping cart robot is a self-propelled vehicle developed by open source software ROS, in which the automatic driving is achieved via image recognition. The proposed dynamic route planning genetic algorithm gets the shortest route compared to other methods in limited time. The robot can guide customers to find their interested commodities following the planned route. In conjunction with the vocal service, the robot is able to notify the customers. The customers are allowed to interact with the robot by the APP.

## Conflicts of Interest

The authors declare no conflict of interest.

## Acknowledgment

This research is co-sponsored by MOST 107-2221-E-024-001-MY3, 106-2221-E-182-015-MY3, ITRI, MoE, and Chang Gung Memorial Hospital, Taoyuan.

## References

- [1] What is Duckietown? Online Available: <http://duckietown.mit.edu/>
- [2] L. Paull, J. Tani, H. Ahn, et al, "Duckietown: an open, inexpensive and flexible and capable platform for autonomy education and research," 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 1497-1504.
- [3] J. Tani, L. Paull, M. T. Zuber, D. Rus, J. How, J. Leonard, and A. Censi, "Duckietown: an innovative way to teach autonomy," International Conference EduRobotics 2016, 2016, pp. 104-121.
- [4] Chi-Shian Lin, "Study on map-based indoor mobile robot vision navigation," Master's Thesis of Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, 2009.
- [5] Ö. Ş. Taş, F. Kuhnt, J. M. Zöllner, and C. Stiller, "Functional system architectures towards fully automated driving," 2016 IEEE Intelligent Vehicles Symposium (IV), June 2016, pp.304-309.
- [6] A. Hellmund, S. Wirges, Ö. Ş. Taş, C. Bandera, and N. O. Salscheider "Robot operating system: a modular software framework for automated driving," 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016.
- [7] Inside Amazon Go, a Store of the Future Online Available: <https://www.nytimes.com/2018/01/21/technology/inside-amazon-go-a-store-of-the-future.html>
- [8] D. Grewal, A. L. Roggeveen, and J. Nordfält, "The future of retailing," Journal of Retailing, vol. 93, no. 1, pp. 1-6, 2017.
- [9] M. R. Garey and D. S. Johnson, "Computers and intractability: a guide to the theory of NP-completeness," W. H. Freeman and Company, San Francisco, 1979.



Copyright© by the authors. Licensee TAETI, Taiwan. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY-NC) license (<https://creativecommons.org/licenses/by-nc/4.0/>).