ON HEARING YOUR POSITION THROUGH LIGHT FOR MOBILE ROBOT INDOOR NAVIGATION

Anonymous ICME submission

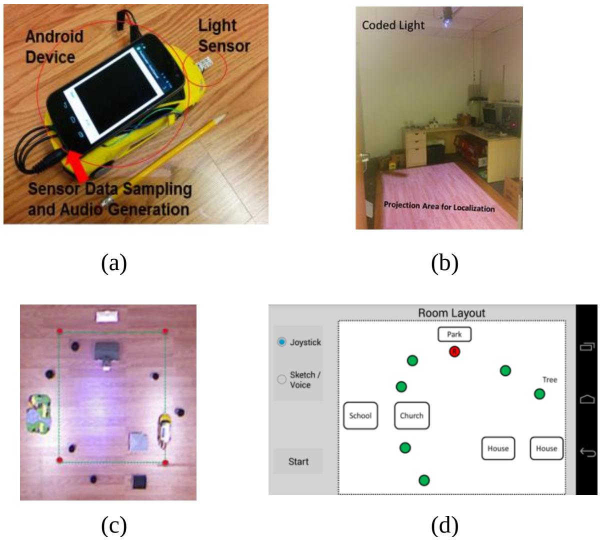
ABSTRACT

Mobile Audio Commander (MAC) is a mobile phone-based multimedia sensing system that facilitates the introduction of extra sensors to existing mobile robots for advanced capabilities. In this paper, we use MAC to introduce an accurate indoor positioning sensor to a robot to facilitate its indoor navigation. More specifically, we use a projector to send out position ID through light signal, use a light sensor and the audio channel on a mobile phone to decode the position ID, and send navigation commands to a target robot through audio output. With this setup, our system can simplify user's robot navigation. Users can define a robot navigation path on a phone, and our system will compare the navigation path with its accurate location sensor inputs and generate analog line-following signal, collision avoidance signal, and analog angular signal to adjust the robot's straight movements and turns. This paper describes two examples of using MAC and a positioning system to enable complicated robot navigation with proper user interface design, external circuit design and real sensor installations on existing robots.

***Index Terms*—** One, two, three, four, five

1. INTRODUCTION

Using state-of-the-art sensors to upgrade existing robots is always a time-consuming and costly process. Frequently, robot owners have to ship their existing robots back to the factory for a firmware update or even hardware upgrade. On the other hand, mobile devices such as Android phones have become ubiquitous in modern society. Considerable research over the past decade has transformed the mobile phone into a platform that supports everywhere sensing, communication, and interaction enabling a variety of applications. Since mobile phone software is easy to be upgraded and the phone itself has many advanced sensors, user interfaces, as well as audio input and output, we think it is a great platform for introducing extra sensors to existing robots. Also because navigation is one of the most important features of robots, we select our first goal as introducing accurate position sensors to existing robots through a mobile phone to achieve high-accuracy interactive indoor navigation.



(a) (b)

(c) (d)

Fig.1. (a) An MAC-enabled robot car; (b) Coded light setup for indoor localization; (c) a robot is navigated on a track; (d) the user client for MAC.

A common method to implement an interactive robot navigation system is to build a customized full-function control unit that can handle real-time robot control, low-level robot IO, networking, GUI, localization, etc. Examples are [7] and [8]. However, these systems are very expensive and may require full recall of all existing robots. Manikandan et al. [4] alleviate this issue by sending cellphone commands to robot's microcontroller through modulated audio outputs, but still require audio input and command decoding implementation on the microcontroller side. Sato et al. [6] overcome these problems by using a general PC as a robot control unit. However, this type of control system requires very careful considerations on the control action latency management.

There are also systems that use cellphones or tablets to obtain users' robot control inputs and use Wi-Fi or Bluetooth to send high-level commands to the robot. Ayres et al. [1] and Gupta [3] use Wi-Fi connection to send commands such as forward, backward, left-turn, right-turn, stop, etc. Since the Wi-Fi system has a big network buffer and may be delayed irregularly because of network traffic, the user experience of this kind of system cannot be very smooth. Other systems such as Moeckel [5] use a more responsive wireless communication form such as Bluetooth to connect a cellphone with a robot. However, using very frequent small Bluetooth packets to transmit many real-time control commands is not a reliable and smooth way to control a robot.

To overcome these problems, we use a mobile phone-based Mobile Audio Commander (MAC) that collects user input, interfaces with a location sensor, decodes location signal, compares decoded location with user defined paths and sends out robot guidance signals similar to robots' existing sensor outputs through audio signal. Unlike camera-based approach, MAC incorporates users into a refined robot control loop seamlessly and takes advantage of their intelligence and knowledge of the environment, thus does not require heavy computation. Unlike existing reactive robots, MAC uses inbuilt phone jack to mimic expected signals for the robot, so that different sensors can be added to robots without customized interfaces. Moreover, MAC has global location information for all objects in the environment. This allows it to navigate the robot in a more natural and precise way. Figure 1(a) shows a car-like robot enhanced with MAC. It is being directed by a user to follow a square track with a light sensor for indoor localization. Figure 1(c) and 1(d) show that several obstacles have been added to the environment. To avoid obstacles, the robot indicated by a red circle has to move along a rectangle path defined by a user. And if any obstacle blocks the defined path, MAC will automatically generate signals to stop the car to avoid collisions.

In the next section, we explain the requirements for MAC construction. We then describe the MAC system and its modules in details, which are an interaction of both hardware and software. Several applications are also demonstrated followed by evaluations. Finally, we end with conclusions and discussions for future work.

2. MAC ASSUMPTIONS

MAC is a mobile phone-based sensing platform to introduce extra sensors and natural user interface on existing robots. Given the diversity of robots on today's market, we make the following assumptions while designing MAC:

1. The target robot can be driven differentially. As the most common control mechanism for robot navigation, differential driving has been used extensively for existing robots. MAC is designed to enable sensing capability on differential wheeled robots with two independently driven wheels fixed on a common horizontal axis or three wheels where two independently driven wheels and a roller ball or a castor attached to maintain equilibrium.
2. Analog-to-digital (AD) based communication channels are available for these robots. MAC simulates output signals of existing sensors and feeds them into robots' control units by sending simulated analog signals to their AD pins.

3. MAC PLATFORM

3.1. System Overview

MAC is a generic platform designed for inclusion of advanced sensing capability into existing robots by putting together insights from the fields of ambient intelligence and robotics. In this approach, advanced robotics functionalities are not achieved through the development of extremely complex robots with various customized sensing devices, communication channels, and computation units, but through the cooperation of existing components, including necessary sensors, control units of robots, inbuilt audio channels and computation units on mobile devices.

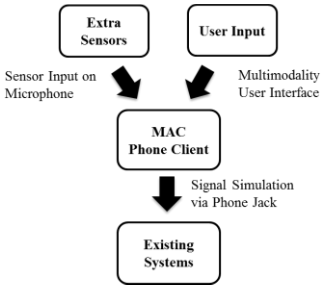

Extra Sensors
User Input
Sensor Input on Microphone
Multimodality User Interface
MAC Phone Client
Signal Simulation via Phone Jack
Existing Systems

Fig.2. MAC Architecture.

At the core of MAC's functionality, there are three critical components, sensor output sampling on the microphone, a user client running on the phone, and simulated signal generation via the phone jack, as shown in Figure 2. Through the microphone, MAC can collect and decode data output from different types of sensors. By running a phone application, MAC considers users as one special type of sensors and provides rich user interfaces for natural interaction so that it can leverage users' knowledge of the environment. This scheme relieves MAC from heavy computation for certain tasks such as object recognition. Finally, generating simulated signals in software and sending them through phone jack allows MAC to communicate with existing robots without modifying their hardware connections.

3.2 Data Transmission on Phone's Audio Jack

There are multiple advantages of using the audio jack for data collection. Firstly, it is a standard interface across phones from all manufacturers, and therefore, can be used as a generic data transport. Secondly, as a wired analog communication channel, audio jack can be used for instant data transmission. More specifically, a data sampling rate as high as 44100Hz on the microphone and high quality of sound generation with frequency up to 24000Hz are supported on the Android platform. This means that by connecting the output of extra sensors to the microphone and sending out simulated signals to target robots through sound generation, MAC can talk to both sensors and robots instantly without adding new hardware components.

3.1.1. Sensor Input on Smartphone's Microphone

Microphones on mobile devices provide a one-way analog communication channel for sensor data collection. Output signals of extra sensors can be sampled and processed into digital data at 44.1 kHz. More specifically, for sensors that only need a small bias voltage and offer a transient response, their output can be connected directly between the microphone and the ground line in the audio jack. Then an application running on the phone can sample the voltage on the microphone line and process it for further usage. So far, two types of sensors have been integrated with MAC in our preliminary experiment by connecting their outputs to the microphone. These sensors are an infrared proximity sensor, GP2Y0A21YK0F, and an ultrasonic distance sensor, HC-SR04. Both are popular components in the robotics field.

The output of selected infrared sensor is an analog voltage that is relative to the distance between the sensor and objects in front of it. This output voltage is inversely proportional to this distance and usually in the range of 0 ~ 5 Volt. The output of ultrasonic distance sensor is different. When this sensor detects ultrasonic reflection from objects ahead, it will set one of its pins to high voltage, which usually means 5 Volt, and delay for a period of time. The length of this period is proportional to the distance between the sensor and its obstacles. Both amplitude and duration of the output signals can be calculated by sampling the voltage on the audio input hardware.

3.2.2 Data Transmission using Sound

Android platform also supports writing a continuous stream of sound data to its phone jack for playback, providing a communication channel from the phone to existing systems that MAC can use for sending real-time driving commands such as going forward, turning left/right, and decreasing speed.

In our current implementation, communication data is interpreted as the amplitude of sine waves. This means that for different commands, MAC will generate a period of sine wave in real-time with different amplitudes. A standard circuit for rectification and signal conditioning is connected to the output signal to create corresponding direct current (DC) voltage, which will be collected by the control unit on the robot. Using this scheme, MAC can mimic various control signals from different types of sensors in software and feed them into existing systems with expected format without adding dedicated hardware connections. For example, in our demonstration of robot navigation, we have successfully generated simulated line-following signal, obstacle avoidance signal, and rotation signal and fed them into the analog-to-digital converter (ADC) on Arduino-based robots. On the other side of this conversation, robots can restore original commands by sampling these fed analog signals. These signals are mimicking the output of extra sensors that do not exist on original robots, such as the infrared proximity sensor and the ultrasonic distance sensor that we discussed above and can be used to drive robots to follow a specific path, avoid obstacles or make turns.

4. MAC APPLICATIONS

To demonstrate how MAC can be used to enhance existing robots with advanced sensing capability and facilitate robot navigation experience, we have built two prototypes of intelligent robot cars equipped with MAC, shown in Figure 1(a) and Figure 4(a) respectively. An Android application has also been developed as the MAC user client and provides rich interfaces for users to drive these cars to perform different tasks. We also use coded light as our positioning system so that the location information of our robots and obstacles in the environment can be obtained.

4.1 Coded Light for Indoor Localization

Fan et al. [2] explored the feasibility of using coded light to attach digital information to arbitrary physical objects and retrieve the attached media later based on the phone's position. For that purpose, a sequence of gray code images was projected onto target surfaces and a light sensor was used to sample the light level at points of interest. A decoding algorithm was then used to restore the sensor's position. In their implementation, a high localization speed of 80 times per second was achieved. We develop our own coded light-based indoor localization system as shown in Figure 1(b). Different from Fan's system, we mount our projector on the ceiling and use it to project the coded light onto the floor. This setup allows us to locate an existing robot 80 times per second within an area of 2.4m×1.3m by attaching a light sensor to it. In our system, the data output of the light sensor will be collected by MAC user client from a microphone and decoded to obtain the robot's real-time position. Given the size of the projection area and the maximum number of points the system can differentiate, a resolution of 3 square meters is achieved.

4.2 Multimodality User Interface

Users may vary in the favor of how to interact with their robots. For this navigation demonstration, three user interfaces have also been developed to demonstrate how MAC can be used to improve user's navigation experience with existing robots by taking advantage of the rich interaction modalities on modern mobile phones.

From a preliminary study, we learn that some users prefer direct control of the robot's motors so that they can manually move a robot in a certain direction with a joystick-like control panel (see Figure 3(a)). However, some of our users may suffer from limited dexterity or muscular dystrophy, and fine motor control can be difficult. For example, to direct a mobile robot through the path between two tables that is barely wider than the robot itself may require too much effort.

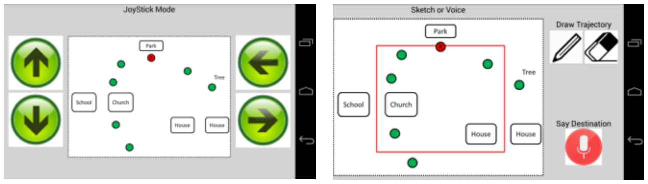


Fig.3. (a) Direct control mode; (b) Sketch and voice mode.

Therefore, MAC also allows users to draw an approximate path on the touchscreen with fingers to drive a robot (see Figure 3(b)). This path would be described as a string of positions that can be used to guide the robot to follow lines, avoid obstacles, and make necessary turns. Users can also give high-level commands such as "John's place" or "the church" by speaking to the phone. This feature is achieved by assigning proper names to potential targets and collecting their positions in the environment beforehand. When the user wants to direct the robot to one of the places, he can invoke the inbuilt voice recognition service in MAC by pressing the voice button and speak out the name of that place. MAC will retrieve the target's position automatically, and generate a desirable path from the robot's current position to the retrieved location.

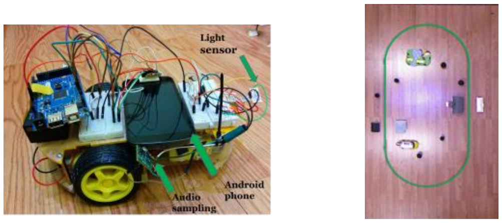
4.2.1 Direct Control

Figure 3(a) shows the direct control mode in MAC. Similar to a physical joystick, users can control the rotation direction of drive wheels on the robot separately by pressing different buttons, Forward, Backward, Left, and Right. A layout of the room is also displayed in the middle of the display showing where all obstacles are, denoted by green dots. The red dot on the map indicates the current position of the robot. The location information for all obstacles is collected beforehand by putting light sensors on top of them and saved in the database.

4.2.2 Sketching and Voice Control

Users can designate a specific route to the robot by drawing freestyle paths on the touch screen as well. Figure 3(b) exemplifies such a case, in which a user draws four trajectories to form a square path around the whole scene. The robot, which is represented by a red dot, is following the defined paths. Before approaching corners, the robot will decrease its speed gradually, make 90-degree turns, and then keep moving straight.

Voice commands are also available by simply pressing the voice button to invoke voice navigation service. Users speak a place's name out, for example, "John's house," and MAC will recognize this place by comparing it with predefined places in the database and obtain its location in the room. After this, a set of waypoints will be generated to represent the potential path between the robot's current location and the expected destination and navigate the robot to get there. This functionality is based on the voice recognition service on Android platform.



(a) (b)

Fig.4. (a) First MAC prototype based on Arduino car; (b) An oval-shaped track for car navigation.

4.3 First Prototype with Oval-shaped Track

Our first prototype is based on an Arduino car-like robot, and a Samsung Galaxy Nexus Android device is used to run MAC and collect location signal from the light sensor (see Figure 4(a)). With the setup of coded light, MAC can sample light strength signal from its microphone and obtain the robot's real-time location. By comparing the location data from the light sensor and users' expected path, MAC would calculate the difference between them and send various commands to adjust the robot's motion to follow an oval-shaped track (see Figure 4(b)) by streaming sine wave signals via the audio jack. Available commands in our current implementation include: going forward, decreasing speed, turning right, and turning left. As discussed in section 3.2.2, these commands would be translated into sine waves with corresponding amplitudes and sent to AD pin of the Arduino board on the robot. As the control unit of the robot, the Arduino board keeps sampling the voltage level on its AD pin to restore original commands.

While the robot is moving forward, a PID controller implemented on the robot is used to adjust the speeds of both its left and right wheels minimizing the distance between its position and the expected path. If obstacles appear ahead, a second PID controller is used to decrease the robot's speed based on the distance from the robot to the obstacle without running into it.

4.4 Second Prototype with Sharp Turns

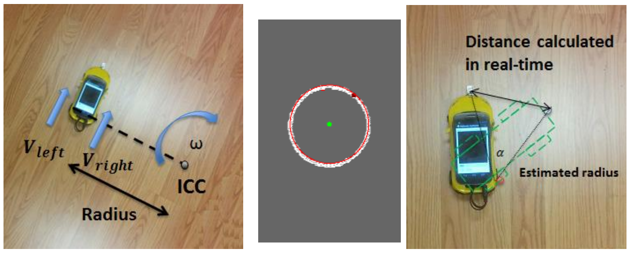
Our second robot is based on an off-the-shelf Lamborghini-like robot (see Figure 1(a)). The same Android device and light sensor are used to run MAC and collect the received light intensity from its microphone. Similar to our first prototype, two PID controllers are used to adjust the speeds of two drive wheels. Different from our first prototype, its second PID controller would decrease the robot's speed gradually while approaching corners denoted as red dots in Figure 1(d). Moreover, this prototype also implements 90-degree point turns with the help of a third PID controller. Such capability enables robots to take sharp turns in very less space.

4.4.1 Rotation Radius Estimation

As the differential drive mechanism describes, robots can perform rolling motion if different velocities are applied to the wheels. In such case, robots will rotate about Instantaneous Center of Curvature, which is known as the ICC (see Figure 5(a)). In our current implementation, the robot has been rotated about one of its back wheels, left wheel for turning left and right wheel for turning right (see Figure 5(c)). However, both the position of rotation point and radius change dynamically as wheel slip is inevitable during rotation. In particular, while rotating, the robot forms an under-actuated nonlinear dynamic system where neither lateral nor longitudinal slips can be ignored. To investigate the maneuverability of such a system in the realistic environment, we propose a solution with the following steps to find the "real" rotation center and radius:

1. Put the robot at several places in the environment and perform 360-degree rotation for certain times
2. Record the location data of the light sensor installed on the robot at high frequency
3. Find the average center and radius with minimum errors based on the recorded positions

Figure 5(b) shows one of the results from our rotation experiments. The white dots indicate all the collected positions from the light sensor while the robot is doing the 360-degree rotation for four times. The estimated instantaneous center of curvature has been calculated by averaging all received locations. We also estimate the rotation radius by averaging the distances from each sensor location to the estimated center. At this particular location, we find an average radius of 155mm, given that the distance from the light sensor to one of the wheels is 150mm. The estimated rotation center is denoted by the green dot. We also plot out the resulting rotation track, which is the one in red. By collecting the average radius at different locations, we can calculate the one with a minimum error for the whole scene and this radius will be used for point turn PID control as described below.



(a) (b) (c)

Fig. 5. (a) Rotation mechanism for differential driven robots; (b) Radius estimation for point turn; (c) PID control for point turn with turning angle as input.

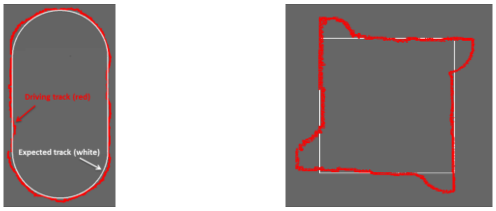
4.4.2 PID Control for Point Turn

While following a track like the one in Figure 6(b), robots need to perform rotations for a certain degree at turning points. In the second prototype, a third PID controller is developed to control the rotation with high accuracy. Take turning right as an example. In the beginning, the received position from the light sensor will be recorded and saved. Then the robot's right rear wheel will be held still while the left wheel would be driven forward. This means that the rotation center would be around the right wheel. While the robot is making the turn, the distance between the saved location and the light sensor's real-time location will be calculated 80 times per second. By assuming that the robot is rotated consistently with the estimated radius discussed above, an isosceles triangle can be formed as shown in Figure 5(c), where α indicates the turning angle. This angle will be further used as input of the third PID controller to control how fast the rotation will be performed until the expected rotation angle is achieved.

5. MAC EVALUATION

We conduct an early stage performance evaluation of MAC design. In this evaluation, two tasks have been assigned to our two prototypes respectively. The first prototype is navigated through an oval-shaped track (see Figure 4(b)) on the floor with coded light for navigation. The result of this evaluation is shown in Figure 6(a). The white line is the expected track that the robot should follow and the red line, which is formed by all the locations collected from the light sensor on the front of the robot, indicates the real driving path of the robot. By comparing the difference of expected path and real track, we obtain the following result. The maximum error of the navigation for our first prototype is 4.3cm and the minimum error is 0. Through the whole track, a mean error of 1.4cm is reported.

One goal of MAC is to navigate users' existing robots in a realistic environment, such as living rooms or bedrooms. For that purpose, MAC must be aware of all the obstacles in the environment and drive the robot to avoid them. Moreover, some users may have limited space in their house, and this requires the robot to make sharp turns in a small space. Therefore, we conduct another evaluation where the second prototype was driven through a square track (see Figure 1(d)). On this track, there are also four turning points at which the robot would make 90-degree turns. Moreover, before approaching these turning points, the robot also needs to decrease its speed gradually.



(a) (b)

Fig.6. (a) Driving result for oval-shaped path; (b) Driving result for rectangle path.

The result of our second experiment can be found in Figure 6(b), where the robot is navigated to finish a square track once. Similar to Figure 6(a), the white line defines the expected path for the navigation while all reported locations of the light sensor form the real driving path, which is drawn in red. Through the whole travel, 3547 locations are reported, 2863 of which are collected when the robot was moving straight forward with a mean error of 13mm. The maximum error is 41mm, and the minimum is 0. From Figure 6(b), one may also find that four quadrants are developed when the robot performed 90-degree turns at the four corners of the track. 684 positions are obtained during these rotations.

6. CONCLUSION AND FUTURE WORK

This paper describes the use of Mobile Audio Commander (MAC), with which external sensor can be used on existing robots for advanced features, such as environment sensing and indoor navigation. To facilitate the use of MAC, we introduce the design of using the microphone on a mobile device to collect various sensor outputs and processing these data with an application running on the device. We also demonstrate how to use audio generation on an Android phone to mimic the expected sensor inputs for existing robots. More specifically, three types of sensors have been discussed, and the output of them can be simulated using MAC. We also connect MAC to coded light indoor localization system and achieve a high-resolution indoor position for mobile robots. Additionally, we build two prototypes equipped with MAC and introduce three different interfaces, virtual joystick, sketch, and voice commands, to help users navigate their robots indoor easily.

MAC can be extended for more applications in the future. Currently, we only integrate location sensors with MAC for navigation purpose. We also want to look into more types of sensors, such as temperature sensor, airflow sensor, and other environment sensors, and investigate how they can be used to enhance existing robots with more advanced features. Moreover, prior research has shown that the localization method we use can track multiple objects simultaneously; therefore, multi-robots collaboration could be one direction to explore. Finally, with the ongoing popularity of Internet of Things (IoT), more and more objects are integrated to interact with each other. With MAC, the inbuilt networking capability of smartphones can be used to connect existing robots and other IoT devices together to establish a highly networked future. This could help us discover more interesting research topics.

7. REFERENCES

[1] T. Ayres and B. Nolan, "Voice activated command and control with speech recognition over WiFi," *Science of Computer Programming*, vol. 59, pp. 109-126, Jan. 2006.

[2] M. Fan, Q. Liu, H. Tang and P. Chiu, "HiFi: hide and find digital content associated with physical objects via coded light," ACM HotMobile 2014, pp. 6, Feb. 2014.

[3] G. S. Gupta, S. C. Mukhopadhyay and M. Finnie, "WiFi-based control of a robotic arm with remote vision," IEEE I2MTC 2009, pp. 557-562, May. 2009.

[4] D. Manikandan, P. Pareek, and P. Ramesh, "Cell phone operated robot," IEEE INTERACT 2010, pp. 183-184, Dec. 2010.

[5] R. Moeckel, C. Jaquier, K. Drapel, E. Dittrich, A. Upegui, and A. Jan Ijspeert, "Exploring adaptive locomotion with YaMoR, a novel autonomous modular robot with Bluetooth interface," *Industrial Robot: An International Journal*, vol. 33, pp. 285-290, 2006.

[6] N. Sato, K. Kon and F. Matsuno, "Navigation interface for multiple autonomous mobile robots with grouping function," IEEE SSRR 2011, pp. 32-37, Nov. 2011.

[7] I. Wieser, A. V. Ruiz, M. Frassl, M. Angermann, J. Mueller and M. Lichtenstern, "Autonomous robotic SLAM-based indoor navigation for high resolution sampling with complete coverage," IEEE PLANS 2014, pp. 945-951, May. 2014.

[8] H. A. Yanco, "Wheelesley: A robotic wheelchair system: Indoor navigation and user interface," *Assistive technology and artificial intelligence*, vol. 1458, pp. 256-268, May 2006.