201 | C. Qian, D. Page, A. Koschan, and M. Abidi, "A 'Brick'-Architecture-Based Mobile Under-Vehicle Inspection System," in Proc. of SPIE Unmanned Ground Vehicle Technology VII, Vol. 5804, Orlando, FL, pp. 182-190, March 2005

# A “Brick”-architecture-based mobile under-vehicle inspection system

Cheng Qian[1], David Page, Andreas Koschan, Mongi Abidi

Imaging, Robotics, and Intelligent Systems Laboratory
Department of Electrical & Computer Engineering
The University of Tennessee
Knoxville, TN 37996

## ABSTRACT

In this paper, a mobile scanning system for real-time under-vehicle inspection is presented, which is founded on a “Brick” architecture. In this “Brick” architecture, the inspection system is basically decomposed into bricks of three kinds: sensing, mobility, and computing. These bricks are physically and logically independent and communicate with each other by wireless communication. Each brick is mainly composed by five modules: data acquisition, data processing, data transmission, power, and self-management. These five modules can be further decomposed into sub-modules where the function and the interface are well-defined. Based on this architecture, the system is built by four bricks: two sensing bricks consisting of a range scanner and a line CCD, one mobility brick, and one computing brick. The sensing bricks capture geometric data and texture data of the under-vehicle scene, while the mobility brick provides positioning data along the motion path. Data of these three modalities are transmitted to the computing brick where they are fused and reconstruct a 3D under-vehicle model for visualization and danger inspection. This system has been successfully used in several military applications and proved to be an effective safer method for national security.

**Keywords:** Modular architecture, multimodal data fusion, navigation planning.

## 1. INTRODUCTION

With national security raised to a major problem in our lives, the research on under-vehicle (UV) inspection has drawn more attention than before. A successful UV inspection system should not only have a low-profile figure to work in the narrow space, but also have an omni-directional motion ability, a multimodal sensing ability, and a fast computation ability that can acquire, process and display the data in real time. This mobility-sensing-computation paradigm enables us to follow a “Brick” concept to design the UV inspection systems.

The motivation of “Brick” concept lies in our wish that one day building an inspection system is as easy as building a computer nowadays. There are various functional bricks such as omni-directional legs, artistic brains, and night vision eyes available to choose, and to build a particular system is just to integrate the bricks with certain functions (See Figure 1). The practice of this concept is based on a generic modular architecture where the system is decomposed into various functional modules with standardized interfaces. Each module should hold a coherent function inside and be decoupled from other modules. Each module should also offer the flexibility of adjusting function specifications and data representation. A system built on this modular architecture will be easy for customization, expansion, and maintenance.

Following this “Brick” architecture, the UV inspection system can be basically decomposed into bricks of three kinds: sensing, mobility, and computing. Each brick is mainly composed by five modules: data acquisition, data processing, data transmission, power, and self-management. These five modules can be further decomposed into sub-modules where the function and the interface are well-defined. In terms of hardware implementation, each brick is built by integrating sensors, single-board computer, wireless card, batteries and some other functional components. These decouplable components are packed into a box to form a physically independent unit, “brick”. These bricks communicate with each other through wireless LAN. In terms of software implementation, we propose a generic model

---

[1] cqian@utk.edu; phone: (865) 974-9737; fax: (865) 974-5459

Unmanned Ground Vehicle Technology VII, edited by Grant R. Gerhart,
Charles M. Shoemaker, Douglas W. Gage, Proceedings of SPIE Vol. 5804
(SPIE, Bellingham, WA, 2005) · 0277-786X/05/$15 · doi: 10.1117/12.602979

for the programs of the modules to follow. This model formalizes the practices of coding the module's main structure, data storage structure, threading, synchronization, and priority functions.

Section 2 briefly discusses the related work in under-vehicle inspection system and modular design for intelligent systems. Section 3 presents a “Brick” architecture for UV inspection systems. Section 4 presents the development of our IRIS UV system by following this “Brick” architecture . Finally the paper is summarized in section 5.

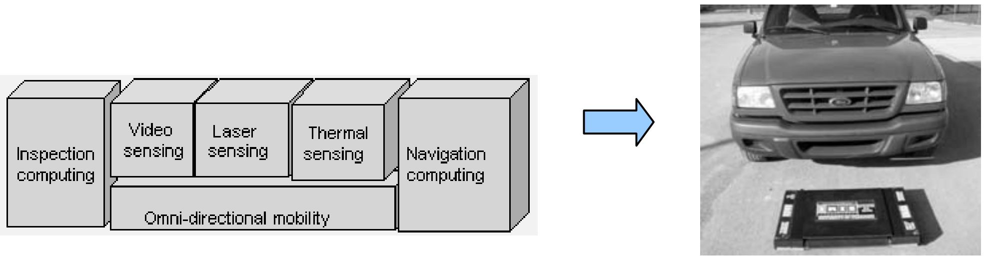


Figure 1. The prototype of the “Brick” concept. A “Brick”-architecture-based under-vehicle inspection system is built by integrating bricks with particular functions.

## 2. RELATED WORK

Starting with a pole-mounted mirror, under-vehicle inspection system has been improved with the technologies in sensors, robotics, and virtual reality. Some commercial systems are the UVIS and ODIS[6,7] by Autonomous Solution, Inc. and UVSS by Northrop Grumman, Inc. These systems carry sensors of IR, sonar, CCD and laser range, and have particular firm motion components to be able to work on rough surfaces. Although these systems have achieved success in military and civil applications, there is still a large space for improvement. First, these systems do not have computation functions. They do not carry automatic navigation capability and each movement is instructed by the operator. Also they do not carry the responsibilities of data analysis and danger inspection, and acquired data are just transmitted to remote computer for computation. Secondly, although modular design is applied to some components, there is no explicit modular architecture to support the entire system design, which brings difficulties in maintenance and expansion of these ad hoc systems.

The “Brick” concept[8,9] proposed by the IRIS lab is about bringing the methodologies in robotics research into the development of practical inspection systems, which has been successfully applied to several imaging and scanning systems developed by this lab. Based on the concept, an inspection system should be divided into physical and logical independent modules, and each module is treated as data processing unit with data input and data output.

Modular design is widely researched in robotics. Jim Albus[1] proposed a grid form architecture, RCS (real time control system) architecture, which aimed to achieve explicit relationship between modules and maximum availability of information to each module. He even made his design the standards in robot industry for a while. His architecture grew through four generations. Fischertechnik, a German company, and AI lab in MIT produced packages of mechanical components and programmable processors for robot building. Fischertechnika and LEGO TC are respectively the names of their products[3]. Their standardization only reaches to the mechanism level, and the part of intelligence computing is left for users.

Suzuki[4] succeeded in using modular design to achieve the “plug-in” in his vision system. The ability of his robot can be expanded by plugging new sensor modules and corresponding behavior and response modules. Bererton [5] built a robot team for surveillance, and each “millibot” is configured from the modular components of sensing, mobility, and processing. According to the particular mission, the components are customized to endow the millibot with particular abilities. These researches are in line with ours but did not deal with the modular design inside the “physical module”.

## 3. A “BRICK” ARCHITETURE

In this “brick” architecture, the UV inspection system is basically decomposed into bricks of three kinds: sensing, mobility and computing. Each brick is mainly composed of five modules: data acquisition, data processing, data transmission, power and self-management. The diagrams of the there bricks are shown in Figure 2-4.

### 3.1. Three Bricks

In the sensing brick, the self-management module is responsible for activating and deactivating other modules and adjusting their statuses. To make the module status dynamically adjustable, the module of data acquisition, data processing, and data transmission are designed with parameterized interface, and their statuses can be dynamically adjusted by the parameters sent from the user or from the computing brick and via plan adjustment sub-module. However, changing the working status of the sensors usually needs hardware interruption and reconfiguration, which is very time-consuming. Therefore, a "virtual hardware control" sub-module is embedded into the data acquisition module, which imitates the hardware adjustment by resampling, cropping, and recalibrating the raw data from the sensors configured at maximum capabilities.

The sensing brick is equal to a logic sensor. "It consists of signal processing from the physical sensors and the software processing needed to extract the percept. It is the functional built block for perception"[2]. The serial processing sub-module improves the data quality by using local straightforward computation methods to fulfill the tasks such as noise filtering, image enhancement, and image merging, whereas the modality conversion sub-module is responsible for extracting the percept of certain modality from the data with the sensor modality.

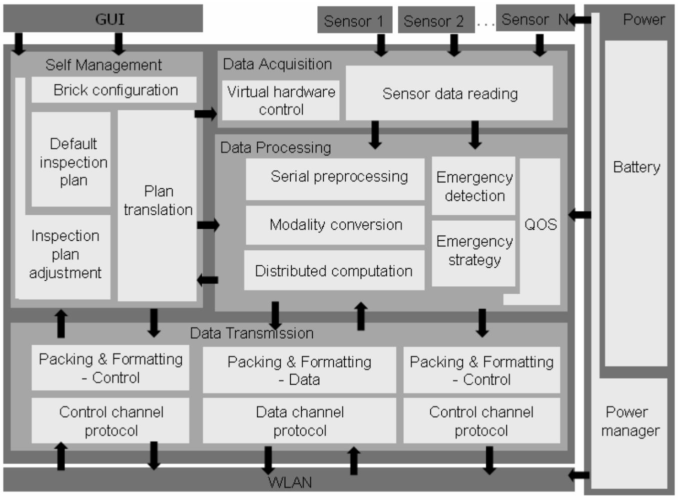


Figure 2. Structure of the sensing brick that contains modules of self-management, data acquisition, data processing, data transmission , and power. Each module is further decomposed into sub-modules. The arrows represent the data or command flow.

In the mobility brick, the planned navigation scheme sub-module has a sequence of goals for the mobility to reach. The goals are fetched out in an optimal order, called exploration strategy. With the reference of the map, a path from the current place to the goal is created. Then based on the path, a sequence of motion vectors are created, which are passed to actuator for execution. At same time, the navigation sensors are acquiring the local environment data. These data are compared with the map in the localization sub-module to see if the mobility has reached the goal and otherwise how much distance still to be covered and how much deviation still to be compensated. All these navigation steps are customized by a navigation plan stored in the self-management.

The motion and localization records are transmitted through the transmission module to other bricks for the use of 3D object reconstruction and deliberative navigation planning. At same time, multimodal data and precise judgment about the environment are transmitted into mobility brick from the sensing and computing bricks, based on which the map are updated, and navigation plan are adjusted.

In the computing brick, we adopt a hybrid paradigm to account for both deliberative and reactive functions required by UV inspection systems. To guarantee low missed alarm rate and false alarm rate, data with high resolution and full coverage over the UV scene should be acquired. These large volume data should be deliberatively processed to find hidden suspicious objects and set alarm with certainty. At same time, mobile UV inspection requires fast reaction towards dangerous situations in inspection and navigation, therefore specific reactive processing modules are embedded. In fact, we distribute the reactive processing over all the bricks, i.e. in the emergency detection and the emergency strategy sub-modules in sensing and mobility bricks. All the reactive responses are transported between bricks through prioritized command channels rather than data channels.

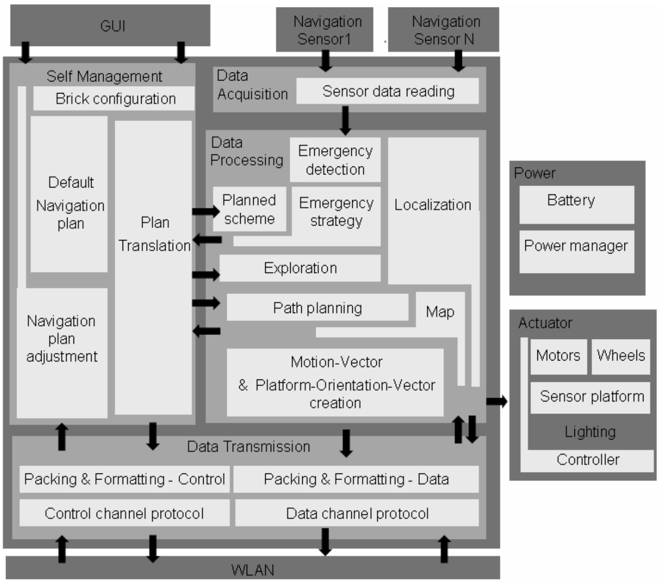


Figure 3. Structure of the mobility brick that contains modules of self-management, data acquisition, data processing, data transmission, actuator and power. Each module is further decomposed into sub-modules. The arrows represent the data or command flow.

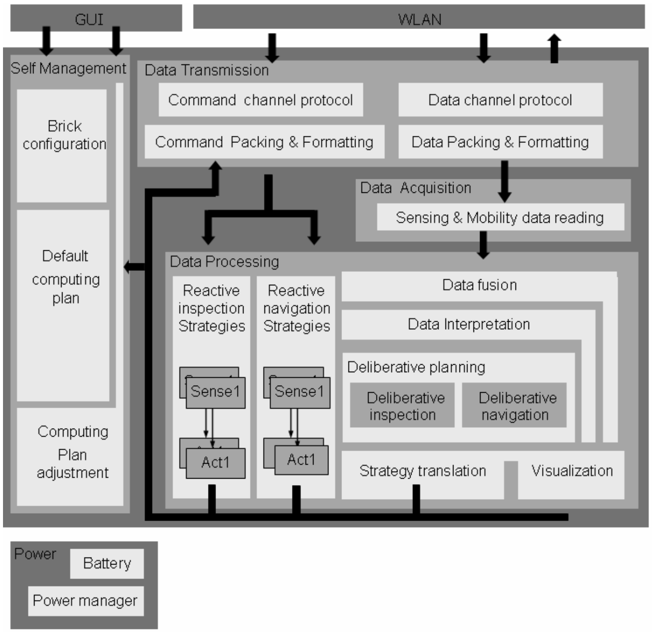


Figure 4. Structure of the computing brick, which contains modules of self-management, data acquisition, data processing, data transmission , and power. Each module is further decomposed into sub-modules. The arrows represent the data or command flow.

### 3.2. Hardware and Software Implementation

Each brick is built by integrating sensors, single-board computer, wireless card, batteries and some other functional components. These decouplable components are packed into a box to form a physically independent brick. The bricks communicate with each other through wireless LAN. Figures 5,6,7 illustrate the hardware decomposition.

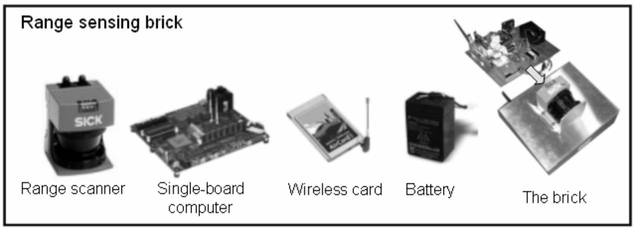


Figure 5. Hardware implementation of a range sensing brick. The range sensing brick is built by integrating a range scanner, single-board computer, wireless card, and batteries.

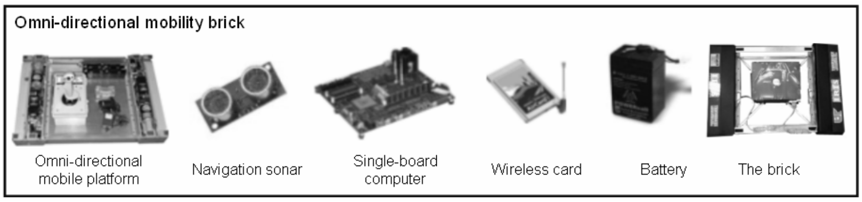


Figure 6. Hardware implementation of an omni-directional mobility brick. The omni-directional mobility brick is built by integrating an omni-directional mobile platform, navigation sonar, single-board computer, wireless card, and batteries.

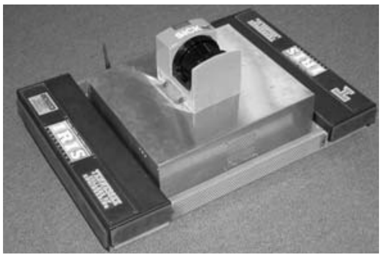


Figure 7. Hardware implementation of an inspection system with a rang sensing brick and an omni-directional mobility brick.

As to those sub-modules implemented by software program, most of them follow the routines of receiving data, processing data, and passing data to modules linked downstream. We call these sub-modules as worker modules and develop a generic model for their program to follow (See Figure 7). This model formalizes the practices of coding the module's main structure, data storage structure, threading, synchronization, and priority functions, but leaves module's processing functions for customization. Based on this model, each worker module is programmed as a single independent thread. Without being subject to a central scheduling module, each thread favors or penalizes its competence for CPU time slices by its "dynamic priority". The priority of a worker module is decided by the suspension status of itself and its linked modules.

We also developed a specific communication DLL for programming the protocol of wireless communication between bricks. It provides standardized functions for creating/terminating the data/command channels and sending/receiving data/commands.

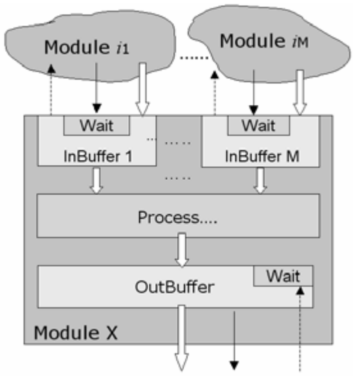


Figure 7. Module program model where the module's main structure, data storage structure, threading, synchronization, and priority functions are formalized.

## 4. IRIS UNDER-VEHICLE INPSECTION SYSTEM

Following this architecture, we built a system composed of four bricks: two sensing bricks consisting of a range scanner and a line CCD, one mobility brick and one computing brick. The sensing bricks capture the geometric data and texture data of the under-vehicle scene, while the mobility brick provides the positioning data along the motion path. Data from all these three modalities are transmitted to the computing brick where they are fused and reconstruct a 3D UV model for visualization and danger inspection.

In the line CCD brick, we use the CCD sensor HRLC CLS-M80250 ( AdaptiveOptics, Inc.), a three-channel line-scanning CCD with a parallel port for data communication and a RS232 serial port for control. This sensor can capture the scene texture with maximum resolution of 10000 pixels/line and maximum speed of 1300lines/sec. This sensor is coupled with "Road Runner-2" (BitFlow Inc.), a PCI-based frame grabber to transport the data from the CCD buffer to the motherboard. Other hardware components in this brick include Mini-ITX Mainboard (VIA Inc.), batteries, DC-DC converter, and WLAN adapter.

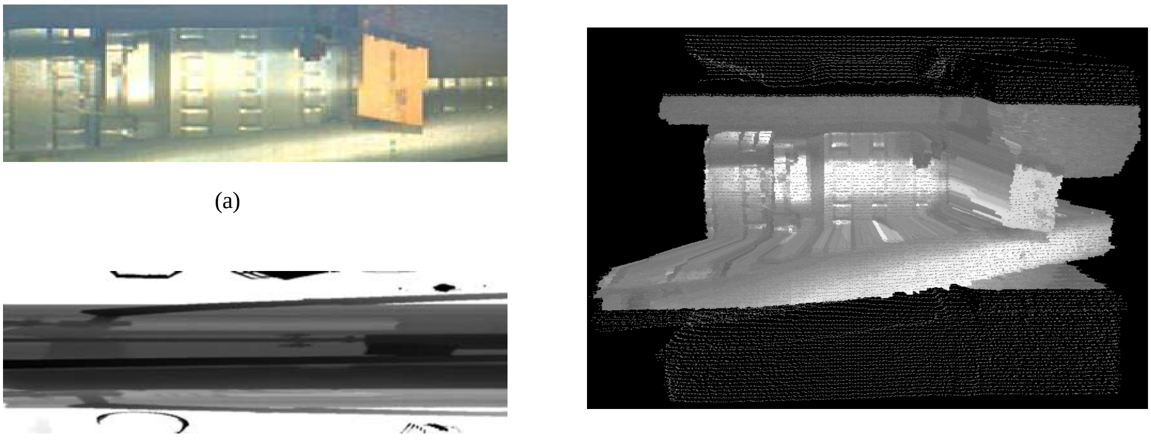
The scanning speed and resolution is adjusted in the virtual hardware control to match the general speed of data flow in this brick. In the serial processing sub-module, straightforward schemes of color calibration and image enhancement are programmed, which require small computation and can be executed in parallel with the data streaming. In the emergency detection sub-module, the image is compared with the color patterns of dangerous objects, and positive results will trigger the emergency strategy sub-module. In the emergency strategy sub-module, the detection results are directly sent to the computing brick by command channel, and a command for increasing the scanning resolution is sent to the virtual hardware control sub-module via the plan translation sub-module to instruct the "sensors" to get more detailed data for detection confirmation.

In the range brick, we use the Sick Laser Measurement System LMS 200[10] as the range sensor. It has the maximum scanning angle of $180^o$, maximum resolution of $0.25^o$, maximum baud rate of 500,000, and the measurement accuracy of $\pm 2cm$. It is communicated with the motherboard by a special high-speed serial interface card ( MOXA,500kBaud). In its serial processing sub-module, a median filter is applied to remove noises and outliers from the raw data. In the modality conversion sub-module, geometric elements such as feature points, curves and surfaces are extracted, which are then packed and transmitted with the range data or transmitted upon request from other bricks. In the emergency detection sub-module, range images are compared with the under-vehicle shape pattern of the automobile and mismatched data are used for danger recognition.

In the mobility brick, we use the SafeBot mobile platform with two independent track modules as the actuator. Each track is controlled by a Robot Controller. This controller chip receives the motion vectors from path planning sub-module and outputs Pulse Width Model (PWM) value to drive the motor. In this chip, interrupts from an optical encoder (Inc US Digital) are counted as moving distance and a PID feedback scheme is embedded for speed stabilization. On this platform a circle of strong LED is installed to illuminate the under vehicle scene.

Another two LMS 200 sensors are placed back to back on the SafeBot as the navigation sensors, which provide a $360^0$ range panorama of the environment. With these range data, a greedy mapping method is implemented for the navigation and path planning in the “unknown” scene under vehicle, and four wheels are used as scene signatures for triangulating the localization. To speed up the movement, a shallow “correction loop” is adopted in the path planning. Therefore the mobility is not required to reach the destination point precisely, but records the deviation vector with the actual position coordinates. The emergency detection sub-module is responsible for detecting the abnormities in the movement that may occur when the mobility is stuck into some places or hit some obstacles. The positive detection results will trigger the emergency strategy sub-module where an intermediary navigation plan will be executed to guide the mobility away from the trouble. The relevant information about the detection results and the intermediary navigation plan will be directly sent to the computing brick through command channel.

The computing brick is customized for the two sensing bricks of line CCD and range and the mobility brick. In the data acquisition module, three data queues are placed to receive texture profiles from the line CCD brick, range profiles from the range brick and motion coordinates sequences from the mobility brick. In the data fusion sub-module, we take advantage of the similarities between the camera models of the two sensings and map the texture profiles to the range profiles with compatible time tags. Inside the profile mapping, the polar coordinates of texture pixels are linearized by a 5-order distortion model [11,12], and the pixels are then evenly filled into the intervals of range points[13]. From the fused images textured meshes are rendered and further approximated by a super-shape representation. The fused images as well as the reconstructed model are showed in a 3D visualization environment where the 3D data can be translated, rotated, and zoomed (See figure 9). To save computation resource, the 3D data are stored in a hierarchical structure for interactive multi-resolution and multi-block visualization and displayed with the resolution and dimension fitting into the visualization window.



(a)

(b)

(c)

Figure 9. Visualization. a) Texture profiles  b) Range profiles c) Fused image displayed in a 3D visualization environment.

In data interpretation sub-module, vehicle components are segmented and recognized, and suspicious objects are detected by matching the fused images with both the texture and the geometry patterns of dangerous objects. Global and iterative scheme with heavy computation is involved in this sub-module. In the deliberative inspection sub-module, a seven-alert-level system is embedded. The alert level is determined by indication strength from the detection results, scanning resolution, object distance and orientation to the sensor. Strong indication, high resolution, small distance, and orientation to the sensor result in high level alert. Once an alert level is set, commands for increasing the scanning resolution or the field of view of the sensor will be sent to the sensing, and commands for moving towards to the suspicious objects or changing the orientation of the senor platform will be sent to the mobility. If the detection results keep positive afterwards, the alert level will be heightened, otherwise the alert will be removed.  Once the highest level is reached, an alarm signal will be set. In the deliberative navigation sub-module, the revision of under-vehicle map is created based on the information from the finely registered data and the interpretation results. This alert system is also applied to the reactive planning module where the corresponding strategy can be directly retrieved by a specific input of detection results or navigation reports.

## 5. CONCLUSIONS

In this paper, we proposed a generic architecture of mobile inspection systems for under-vehicle inspection, which is composed of three kinds of bricks, sensing, mobility, and computing. These bricks are physically and logically independent and communicate with each other by wireless communication. These bricks are further decomposed into modules whose functions and interfaces are defined. We built an IRIS under-vehicle inspection system based on this architecture.

In the future, we will continue this research by targeting the following problems. Firstly, it can be found that too much computation is loaded onto the computing brick. Therefore, in the future the computation tasks will be distributed over the whole architecture, and the sensing and mobility bricks will deliver levels of computed information more than raw data. Secondly, the bricks will have a self-calibration module. Thirdly, this “Brick” prototype will be expanded to a “sensor networking system” that contains various “agents”. Each agent is built by integrating a particular sensing brick and a particular mobility brick. These self-contained agents can join a network by registering to the computing and are responsible for executing the navigation and inspection strategies planned by the computing and reporting relevant data to the computing.

## ACKNOWLEDGEMENTS

We greatly thank Cheng Chang for his work in developing the mobility bricks and navigation algorithms, and Zhenhao Cheng for his work in developing the communication DLL and building the wireless LAN for brick wireless communication.

This work is supported by the University Research Program in Robotics under grant DOE-DE-FG02-86NE37968, by the DOD/RDECOM/NAC/ARC Program, R01-1344-18, by the US Army under grant Army-W56HC2V-04-C-0044, and by FAA/NSSA Program, R01-1344-48/49.

## REFERENCES

1. J. S. Albus, “A reference model architecture for intelligent systems design,” in *An Introduction to Intelligent and Autonomous Control*, Kluwer Academic, Publishers, 1993. pp 27-56.
2. R. Murphy, “Chapter 6: Common Sensing Techniques for Reactive Robots” in *Introduction to AI Robotics*. The MIT Press, 2000.
3. F. G. Martin, *Robotics Exploration: A Hands-On Introduction to Engineering*, Prentice Hall, 2001.
4. K. Suzuki, R. Hikiji and S. Hashimoto, “Development of an Autonomous Humanoid Robot, iSHA, for Harmonized Human-Machine Environment,” Journal of Robotics and Mechatronics, Vol.14, No.5, 2002, pp. 324-332.
5. C. Bererton, L. Navarro-Serment, R. Grabowski, C. Paredis and P. Khosla, “Millibots: Small Distributed Robots for Surveillance and Mapping,” Government Microcircuit Applications Conference, March, 2000.
6. K. Moore, N. Flann, S. Rich, M. Frandsen, Y. C. Chang, J. Martin, M. Davidson, R. Maxfield and C. Wood, “Implementation of an Omni-Directional Robotic Inspection System (ODIS),” Proceedings of SPIE Conference on Unmanned Ground Vehicle Technology III, vol 4364, pp. 200-210, Orlando, FL, April 2001.
7. W. Smuda, P.L. Muench, , and G.R. Gerhart, “Autonomy and manual operation in a small robotic system for under-vehicle inspections at security checkpoints, ” Proc. SPIE Vol. 4715, p. 1-12, Unmanned Ground Vehicle Technology IV, Jul,2002.
8. A. Koschan, D. Page, J.-C. Ng, M. Abidi, D. Gorsich, and G. Gerhart, “SAFER Under Vehicle Inspection through Video Mosaic Building,” International Journal of Industrial Robot, Vol. 31, No. 5, pp. 435-442, September 2004.

9. D.L.Page, Y. Fougerolle, A. Koschan, A. Gribok, M.A. Abidi, D.J. Gorsich, and G.R. Gerhart, “SAFER vehicle inspection: a multimodal robotic sensing platform”, Proceedings of the SPIE Defense & Security Symposium, Unmanned Ground Vehicle Technology, Volume 5422, pp. 549-560, 2004.
10. The Sick Company, “LMS / LMI 400 Definition of Telegrams between the User Interface and LMS or LMI Systems via RS 422/RS 232, Version 05.00,” Product Maunal, 1999.
11. R. Swaminathan and S. K. Nayar. “Non-Metric Calibration of Wide Angle Lenses and Polycameras.” IEEE Computer Society Conference CVPR 99, Vol. 2, pp. 413-419, 1999.
12. C. Kolb, D. Mitchell, and P. Hanrahan, "A Realistic Camera Model for Computer Graphics", Proceedings of SIGGRAPH 1995, pp. 317-324,1995.
13. R. Reulke, and A. Wehr, “Panoramic Mapping using CCD-Line Camera and Laser Scanner with Integrated Position and Orientation System,” Proc. Int. Conf. ``Image and Vision Computing New Zealand", Massey Univ., pages 72--77, 2003.