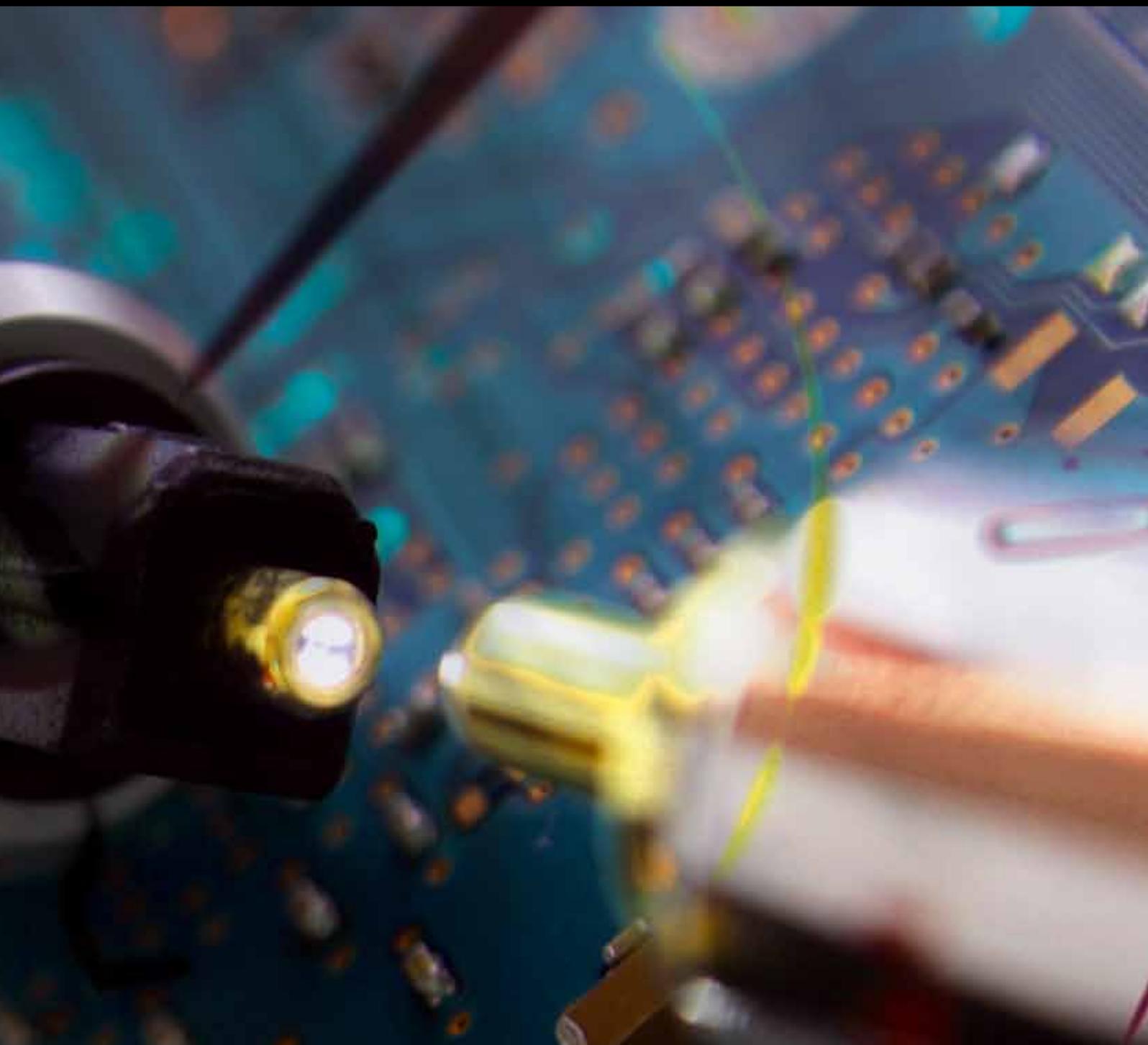


Journal of Sensors

# Sensors for Robotics

Guest Editors: Aiguo Song, Guangming Song,  
Daniela Constantinescu, Lei Wang, and Qunjun Song





---

# **Sensors for Robotics**

Journal of Sensors

---

## **Sensors for Robotics**

Guest Editors: Aiguo Song, Guangming Song,  
Daniela Constantinescu, Lei Wang, and Qunjun Song



---

Copyright © 2013 Hindawi Publishing Corporation. All rights reserved.

This is a special issue published in "Journal of Sensors." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Editorial Board

F. J. Arregui, Spain  
Francesco Baldini, Italy  
B. Bouchikhi, Morocco  
H. Brändle, Switzerland  
Stefania Campopiano, Italy  
Mehmet Can Vuran, USA  
Shoushun Chen, Singapore  
Marco Consales, Italy  
Jesus Corres, Spain  
G. A. Cranch, USA  
Andrea Cusano, Italy  
Dzung V. Dao, Japan  
Cristina E. Davis, USA  
Utkan Demirci, USA  
Xinyong Dong, China  
Denis Donlagic, Slovenia  
C. Dücső, Hungary  
Isabelle Dufour, France  
Luca Francioso, Italy  
Laurent Francis, Belgium

Wei Gao, Japan  
Marco Grassi, Italy  
Tuan Guo, China  
J. Homola, Czech Republic  
Bernhard Jakoby, Austria  
Wieslaw Jakubik, Poland  
K. Kalantar-Zadeh, Australia  
Challa Kumar, USA  
Hiroki Kuwano, Japan  
Yongxiang Li, China  
Yanbiao Liao, China  
Kazusuke Maenaka, Japan  
Eugenio Martinelli, Italy  
Yasuko Y. Maruo, Japan  
Ignacio R. Matias, Spain  
Mike McShane, USA  
Igor L. Medintz, USA  
J. R. Morante, Spain  
Lucia Mosiello, Italy  
M. Nakamura, Japan

K. S. Narayan, India  
Masahiro Ohka, Japan  
Matteo Pardo, Italy  
Michele Penza, Italy  
Ioannis Raptis, Greece  
C. Riziotis, Greece  
G. Sberveglieri, Italy  
M. G. Shlyagin, Mexico  
Pietro Siciliano, Italy  
M. Sriyudthsak, Thailand  
David Stoppa, Italy  
Weilian Su, USA  
Hidekuni Takao, Japan  
Isao Takayanagi, Japan  
Maria Tenje, Sweden  
A. V. Vasilakos, Greece  
Alexander Vergara, USA  
Stanley E. Woodard, USA  
Hai Xiao, USA  
Guangzhong Xie, China

# Contents

**Sensors for Robotics**, Aiguo Song, Guangming Song, Daniela Constantinescu, Lei Wang, and Quanjun Song  
Volume 2013, Article ID 293656, 2 pages

**A Tent Map Based A/D Conversion Circuit for Robot Tactile Sensor**, Jianxin Liu, Xuan Zhang, Zhiming Li, and Xuling Li  
Volume 2013, Article ID 624981, 5 pages

**Rapid 3D Modeling and Parts Recognition on Automotive Vehicles Using a Network of RGB-D Sensors for Robot Guidance**, Alberto Chávez-Aragón, Rizwan Macknoja, Pierre Payeur, and Robert Laganière  
Volume 2013, Article ID 832963, 16 pages

**Dynamic Compensation for Two-Axis Robot Wrist Force Sensors**, Junqing Ma, Aiguo Song, and Dongcheng Pan  
Volume 2013, Article ID 357396, 5 pages

**Modeling a Sensor to Improve Its Efficacy**, Nabin K. Malakar, Daniil Gladkov, and Kevin H. Knuth  
Volume 2013, Article ID 481054, 11 pages

**Feasibility Investigation of Obstacle-Avoiding Sensors Unit without Image Processing**, Yasuhisa Omura, Hiroshi Masuda, and Yoshio Mimura  
Volume 2013, Article ID 643815, 11 pages

**Decision Making in Reinforcement Learning Using a Modified Learning Space Based on the Importance of Sensors**, Yasutaka Kishima, Kentarou Kurashige, and Toshihisa Kimura  
Volume 2013, Article ID 141353, 9 pages

**Multiple Harmonics Fitting Algorithms Applied to Periodic Signals Based on Hilbert-Huang Transform**, Hui Wang, Zhengshi Liu, Bin Zhu, and Quanjun Song  
Volume 2013, Article ID 580152, 9 pages

**Evaluation of Circle Diameter by Distributed Tactile Information in Active Tracing**, Hiroyuki Nakamoto, Futoshi Kobayashi, and Fumio Kojima  
Volume 2013, Article ID 658749, 7 pages

**Force-Sensor-Based Estimation of Needle Tip Deflection in Brachytherapy**, Thomas Lehmann, Mahdi Tavakoli, Nawaid Usmani, and Ronald Sloboda  
Volume 2013, Article ID 263153, 10 pages

**Grey-Level Cooccurrence Matrix Performance Evaluation for Heading Angle Estimation of Moveable Vision System in Static Environment**, Zairulazha Zainal, Rizauddin Ramli, and Mohd Marzuki Mustafa  
Volume 2013, Article ID 624670, 6 pages

## Editorial

# Sensors for Robotics

**Aiguo Song,<sup>1</sup> Guangming Song,<sup>1</sup> Daniela Constantinescu,<sup>2</sup> Lei Wang,<sup>3</sup> and Qianjun Song<sup>4</sup>**

<sup>1</sup> Robot Sensor and Control Lab, School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China

<sup>2</sup> Department of Mechanical Engineering, University of Victoria, P.O. Box 3055 STN CSC, Victoria, British Columbia, Canada V8W 2Y2

<sup>3</sup> School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia

<sup>4</sup> Institute of Intelligent Machine, Chinese Academy of Sciences, Hefei 230031, China

Correspondence should be addressed to Aiguo Song; [a.g.song@seu.edu.cn](mailto:a.g.song@seu.edu.cn)

Received 8 October 2013; Accepted 8 October 2013

Copyright © 2013 Aiguo Song et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Robots are currently widely applied in lots of areas such as industry, agriculture, medicine, transportation, social service, military, space exploration, and undersea exploiting. The robot sensor, as a key component of the robot, has been paid increasing attention by robot researchers. Furthermore, sensor based human-robot interaction is currently the frontier of robot research. Robot sensing technology has been an evolving research field involving a strong interdisciplinary effort from researchers in different fields, namely, electronics, mechanics, material science, measurement and control, signal processing, bioengineering, and so forth. During the last decade, much effort has been done to develop robot sensors for robot control, robot perception, robot navigation, human-robot interaction, teleoperational robot, and so forth. In spite of the large and increasing interest and promising applications, robot sensor design is a significant challenge, which is involved in not only chemical sensors, biosensors, optical sensors, and physical sensors but also signal processing, data fusion, and pattern recognition. For instance, remarkable examples of tactile sensors and systems have been proposed; however, their capability to address specific applications and their extension to other fields such as medical instrumentation, prosthetic devices, and augmented reality are questionable.

This special issue aims at exhibiting the latest research achievement, findings, and ideas in robot sensors. The special issue consists of 10 selected papers summarizing the most recent developments in the field of sensors for robotics. The contents of the special issue cover the main aspects of robot sensing technology such as robot vision, force sensor and

tactile sensor, signal procession of robot sensor, robot sensing and environment modeling, and robot sensor applications.

Robot vision is a fundamental sensation of robot. The paper by Z. Zainal et al. presents a method of extracting information in estimating heading angle of vision system. Integration of grey-level cooccurrence matrix (GLCM) in area of interest selection is carried out to choose a suitable region that is feasible for optical flow generation. From the generated optical flow, heading angle is estimated and enhanced via moving median filter. They demonstrate that GLCM can improve the estimation result of the heading angle of vision system significantly. The paper by A. Chávez-Aragón et al. presents an approach for the automatic detection and fast 3D profiling of lateral body panels of vehicles. They introduce a method to integrate raw streams from depth sensors in the task of 3D profiling and reconstruction and a methodology for the extrinsic calibration of a network of Kinect sensors. This sensing framework is intended for rapidly providing a robot with enough spatial information to interact with automobile panels using various tools. The approach has been applied in the intelligent transportation industry, automated vehicle inspection, automotive production lines, and scan alignment and interpretation.

Force and tactile sensors are absolutely necessary elements for robot when interacting with environment, which call for large range and high accuracy amplifier. The paper by J. Liu et al. developed novel A/D conversion circuits for array tactile sensor by using nonlinear tent map phenomenon, which is characterized by sensitivity to small signal and nonlinear amplifying function. The tent map based A/D

conversion circuit is very suited for multipath signal parallel sampling and A/D converting of large array tactile sensor. The paper by H. Wang et al. describes a new sine fitting algorithm that is able to fit a multiharmonic acquired periodic signal. By means of a “sinusoidal wave” whose amplitude and phase are both transient, the “triangular wave” can be reconstructed on the basis of the Hilbert-Huang transform. This method can be used to test the effective number of bits of A/D converter, avoiding the trouble of selecting initial value of the parameters and working out the nonlinear equations. The paper by J. Ma et al. proposes a dynamic compensation method for improving the dynamic characteristic of two-axis force sensors. The operation frequency of the force sensor system is expanded by a digital filter with backward difference network. Compared to experiment data without compensation, the response time of the dynamic compensated data is reduced by 30%~40%.

Robots rely on sensors to provide them with information about their surroundings. However, high-quality sensors can be extremely expensive and cost prohibitive. Thus, many robotic systems have to use many lower-quality sensors. The paper by N. K. Malakar et al. presents an idea on how modeling a sensor can improve its efficacy when employed within a Bayesian inferential framework. They employ an autonomous robot arm as a test bed, which uses an inexpensive LEGO light sensor to estimate the position and radius of a white circle on a black field. They demonstrate that by incorporating an accurate model of the light sensor spatial sensitivity function (SSF) into the likelihood function of a Bayesian inference engine, an autonomous system can make improved inferences about its surroundings.

Robot sensing and environment modeling are always a key issue for robot navigation and autonomous behaviors. The paper by Y. Omura et al. presents a simple method to detect step height, slope angle, and trench width using four infrared-light-source PSD range sensors. In order to reduce the error of upward slope angle, they also propose a method to improve range-voltage performance of a range sensor. The experiments show that the step height can be measured with high accuracy by using this method. The paper by Malakar et al. proposes a relatively fast learning system in which a robot can autonomously construct a Q-space that considers the importance of the sensors. The paper defines the measure of importance of a sensor for a task, which is the correlation between the value of each sensor and reward in reinforcement learning. Consequently, the state space is reduced based on the importance of the sensors, thus allowing the robot to learn correspondences efficiently.

In this special issue, readers can also find some papers about the application of robot sensors in some important fields. The paper by T. Lehmann et al. developed a virtual sensor for minimally invasive surgery, which is able to precisely sense the needle tip deflection during needle insertion procedures using a 4 DOF force sensor installed on the needle base. The paper by H. Nakamoto et al. presents an active touch method for robotic hands in order to improve dexterity performance of multifinger robotic hands. They define four requirements of a tactile sensor for active touch and design a distributed tactile sensor model, which is able to measure a

distribution of compressive deformation. They also suggest a measurement process with the sensor model and a synthesis method of distributed deformations. The experiments of a five-finger robotic hand with tactile sensors that traces on the surface of cylindrical objects and evaluates the diameters show that the robotic hand can obtain more information about the diameters by tracing the finger.

## Acknowledgments

We would like to thank all the authors for their valuable contributions and also the reviewers for their critical help necessary to achieve a high level of papers' quality and make thus possible the completion of this special issue. Also, we would like to thank all members of the Editorial Board for approving this special issue.

*Aiguo Song*  
*Guangming Song*  
*Daniela Constantinescu*  
*Lei Wang*  
*Qunjun Song*

## Research Article

# A Tent Map Based $A/D$ Conversion Circuit for Robot Tactile Sensor

Jianxin Liu,<sup>1</sup> Xuan Zhang,<sup>2</sup> Zhiming Li,<sup>2</sup> and Xuling Li<sup>2</sup>

<sup>1</sup>NARI Technology Development Limited Company, Nanjing 210061, China

<sup>2</sup>Experiment & Verification Center, State Grid Electric Power Research Institute, Nanjing 210061, China

Correspondence should be addressed to Xuan Zhang; sherry.zhangxuan@gmail.com

Received 10 July 2013; Accepted 25 July 2013

Academic Editor: Aiguo Song

Copyright © 2013 Jianxin Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Force and tactile sensors are basic elements for robot perception and control, which call for large range and high-accuracy amplifier. In this paper, a novel  $A/D$  conversion circuit for array tactile sensor is proposed by using nonlinear tent map phenomenon, which is characterized by sensitivity to small signal and nonlinear amplifying function. The tent map based  $A/D$  conversion circuits can simultaneously realize amplifying and  $A/D$  converting functions. The proposed circuit is not only simple but also easy to integrate and produce. It is very suited for multipath signal parallel sampling and  $A/D$  converting of large array tactile sensor.

## 1. Introduction

In recent decades, with the rapidly development of robot technology, robot sensors have received much attention as a sensing element for robot. Multiaxis force sensors and array tactile sensors, usually called haptic sensors, especially, have become the major research content in the robot sensor research areas [1, 2]. People hope that robot haptic sensor can be like human perception organs which have high measurement accuracy, with similar hand force and tactile organ of integration, miniaturization, and flexibility characteristics. For instance, Song developed a small four-degree-of-freedom wrist force sensor with high precision, which consists of small cross-elastic beam, compliant beams, and the base of the elastic body. It is a kind of self-decoupling force sensor in mechanical structure [3]. Beyeler et al. designed a six-axis MEMS force sensor with a movable body suspended by flexures which allow deflections and rotations along the  $x$ -,  $y$ -, and  $z$ -axes. And the orientation of this movable body is sensed by seven capacitors based on transverse sensing, resulting in a high sensitivity [4]. Ma et al. proposed a novel nonlinear static decoupling algorithm based on the establishment of a coupling error model for 3-axis force sensor in order to avoid overfitting and minimize the negative effect of random noises in calibration data, which can obtain

high precise measurement results of 3-axis force for robot force control [5]. Although robot array tactile sensor can be regarded as a multipoint integrated force sensor, due to flexible and miniaturization requirements of tactile sensor which are high, the measurement principle is more complex than the multiaxis force sensor [6, 7]. Song et al. proposed a novel design of a haptic texture sensor by using PVDF film to fabricate a high-accuracy, high-speed-response texture sensor [8]. Lee and Won developed a novel tactile imaging sensor by using a multilayer polydimethylsiloxane optical waveguide as the sensing probe, which is capable of measuring the elasticity of the touched object with high precision [9]. Based on semiconductor technique, piezoresistive, capacitive, piezoelectric, and other types of robot, array tactile sensors are developing rapidly. Array tactile sensors have a corresponding increase in array size and resolution [10, 11]. Nonetheless, because of measurement mechanism limit, output signal of sensing element of most high-resolution tactile sensors is relatively weak. In addition, as the expansion of tactile array, it hopes that tactile image signal has a fast  $A/D$  conversion rate. Sharing the same  $A/D$  conversion in small scale tactile sensor cannot meet the real-time requirements of signal acquisition for the large scale array tactile sensor.

In recent years, some of the unique properties of nonlinear systems have much in measurement area for their unique

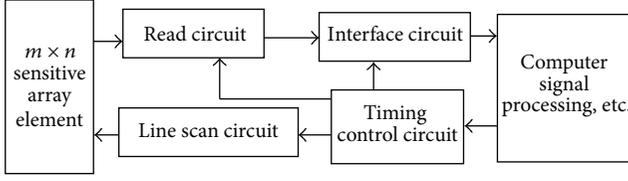


FIGURE 1: Tactile sensor array signal acquisition circuit.

character, such as the chaotic system sensitivity to small signal [12], nonlinear mapping [13], nonlinear information storage [14], and resonance stochastic [15]. In this paper, a tent map is sensitive to small signal circuit and nonlinear transform of unique properties. A novel  $A/D$  conversion circuit for robot tactile sensor array is proposed to achieve parallel sampling of multichannel tactile signals and  $A/D$  conversion with high cost performance, which has a conditioning amplification and  $A/D$  conversion function integration advantages in combination, simple circuit, and easy integration to realize.

## 2. Signal Acquisition System for Robot Tactile Sensor

Typical signal acquisition circuit for robot tactile sensor is shown in Figure 1, including timing control circuit, line scan circuit, read circuit, and interface circuit. The whole signal acquisition process was controlled and coordinated by timing control circuit. According to the arrangements of the timing control circuit, line scan circuit is ordered in  $m$  clock cycles to send the periodic excitation signal to  $m$ -line array sensitive element, while the read circuit is ordered in  $m$  clock cycles to read the output signal of  $n$  column in parallel. Then through interface circuit which consists of signal conditioning and  $A/D$  conversion, tactile signals were transferred to the computer for processing and target recognition.

Conventional robot array tactile sensor, because of the small array size (about  $8 \times 8$ ), often uses an  $A/D$  conversion in order to complete the analog-digital conversion of output signal of  $m \times n$  sensitive elements.

## 3. Small Signal Nonlinear Amplifier and $A/D$ Conversion Based on Tent Map

Tent map was a typical one-dimensional chaotic system [16], which was described as

$$x_{n+1} = T(x_n) = \begin{cases} 2x_n, & 0 < x_n \leq 0.5, \\ 2(1 - x_n), & 0.5 < x_n < 1, \end{cases} \quad (1)$$

where  $x_n \in [0, 1]$ ,  $n = 0, 1, 2, \dots$

This map consisted of two steps: the first step was to uniformly elongate the interval  $[0, 1]$  to its doubled range; the second step was to fold the elongated interval into the original interval  $[0, 1]$ . These iterative operations would cause the separation of adjacent points index, eventually to achieve the state of chaos.

Tent map on the initial value (the system input signal) amplification was different from the linear amplification

method. Linear amplification multiples were a constant and also limited by system operating range. Tent map system in chaotic state amplified the signal doubly and folded the doubled range symmetrically in each iteration, so that the initial small signal could eventually be greatly amplified without beyond range of system operating after several iterations.

The initial value  $x_0$ , which was corresponding to the input signal of tent map system  $V_{in}$ , could be described as a binary fraction

$$x_0 = 0.t_0t_1t_2 \dots = \sum_{j=0}^{\infty} \frac{t_j}{2^{j+1}}. \quad (2)$$

In order to obtain the relationship of the iterative output and the initial signal of discrete tent map, here, this paper would introduce the nonlinear relationship of Bernoulli shift; its kinetic equation was

$$x'_{n+1} = B(x'_n) = \begin{cases} 2x'_n, & 0 < x'_n \leq 0.5, \\ 2x'_n - 1, & 0.5 < x'_n < 1. \end{cases} \quad (3)$$

In each iteration, Bernoulli shift left shifted the binary fraction  $t_1, t_2, \dots$  one place

$$\begin{aligned} x'_1 &= B(x'_0) = 0.t_1t_2t_3 \dots, \\ x'_2 &= B(x'_1) = 0.t_2t_3t_4 \dots. \end{aligned} \quad (4)$$

For Bernoulli shift,  $b_n = \text{sgn}(x'_n - 0.5)$  was defined as the  $n$ th iteration output; there  $b_n = t_n$ , and  $b_i$ ,  $i = 0, 1, 2, \dots$ , was a binary sequence. For tent map, if we define  $g_n = \text{sgn}(x_n - 0.5)$ , then the corresponding relationship between  $g_i$ ,  $i = 0, 1, 2, \dots$ , and  $b_i$ ,  $i = 0, 1, 2, \dots$ , was as follows:

- (1) when  $0 < x_k < 0.25$ , that is,  $b_k = 0$ , then  $B(x_k) < 0.5$ ,  $b_{k+1} = 0$ ,  $T(x_k) < 0.5$ , and  $g_{k+1} = 0$ ;
- (2) when  $0.25 < x_k < 0.5$ , that is,  $b_k = 0$ , then  $B(x_k) > 0.5$ ,  $b_{k+1} = 1$ ,  $T(x_k) > 0.5$ , and  $g_{k+1} = 1$ ;
- (3) when  $0.5 < x_k < 0.75$ , that is,  $b_k = 1$ , then  $B(x_k) < 0.5$ ,  $b_{k+1} = 0$ ,  $T(x_k) > 0.5$ , and  $g_{k+1} = 1$ ;
- (4) when  $0.75 < x_k < 1$ , that is,  $b_k = 1$ , then  $B(x_k) > 0.5$ ,  $b_{k+1} = 1$ ,  $T(x_k) < 0.5$ , and  $g_{k+1} = 0$ .

Therefore,  $g_i$ ,  $i = 0, 1, 2, \dots$ , was a Gray-code sequence of  $b_i$ ,  $i = 0, 1, 2, \dots$ :

$$g_{k+1} = b_k \oplus b_{k+1}, \quad k = 0, 1, 2, \dots \quad (5)$$

According to the above formula and initial time  $x_0 = x'_0 = V_{in}$ , we could obtain

$$\begin{aligned} b_{k+1} &= b_k \oplus g_{k+1}, \quad k = 0, 1, 2, \dots, \\ b_0 &= g_0. \end{aligned} \quad (6)$$

So we designed a tent map iteration output Gray-code sequence  $g_i$ ,  $i = 0, 1, 2, \dots$ , into a binary sequence of Bernoulli

TABLE 1: Results of the A/D conversion.

Sensitive element	(8, 1)	(8, 2)	(8, 3)	(8, 4)	(8, 5)	(8, 6)	(8, 7)	(8, 8)
Output value (mV)	0	0	0	0	25	80	188	246
Gray code	00000000	00000000	00000000	00000000	00000101	00011111	00101000	00100001
$g_0g_1g_2g_3g_4g_5g_6g_7$								
Binary code	00000000	00000000	00000000	00000000	00000110	00010101	00110000	00111110
$b_0b_1b_2b_3b_4b_5b_6b_7$								
Calculated value (mV)	0.0	0.0	0.0	0.0	23.4	82.0	187.5	242.2
Sensitive element	(8, 9)	(8, 10)	(8, 11)	(8, 12)	(8, 13)	(8, 14)	(8, 15)	(8, 16)
Output value (mV)	310	230	170	67	33	0	0	0
Gray code	01101000	00100111	00111110	00011001	00001100	00000000	00000000	00000000
$g_0g_1g_2g_3g_4g_5g_6g_7$								
Binary code	01001111	00111010	00101011	00010001	00001000	00000000	00000000	00000000
$b_0b_1b_2b_3b_4b_5b_6b_7$								
Calculated value (mV)	308.5	226.5	168.0	66.4	31.3	0.0	0.0	0.0

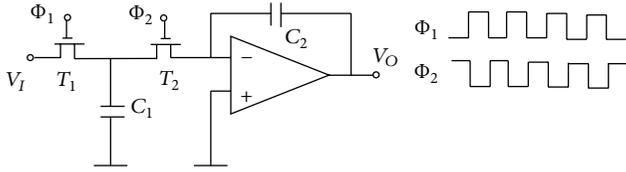


FIGURE 2: Switched capacitor integral circuit.

map  $b_i$ ,  $i = 0, 1, 2, \dots$ , and then calculated the initial value through binary fraction sequence as follows:

$$x_0 = \sum_{j=0}^N \frac{b_j}{2^{j+1}}, \quad (7)$$

where  $g_i$ ,  $i = 0, 1, 2, \dots$ , was the needed digital value. Here, tent map completed signal amplification and A/D conversion function.

#### 4. Implementation on the Circuit of Tent Map for A/D Conversion

A/D circuit basic element was switched capacitor integral circuit shown in Figure 2, where  $T_1$  and  $T_2$  were analog switches and  $C_1$  and  $C_2$  were capacitors. The clocks  $\Phi_1$ ,  $\Phi_2$  were in reverse phase with same period  $T$ . During the former half period of  $T$ , the  $T_1$  was on and  $T_2$  was off; thus the  $C_1$  was charged by input voltage  $V_I$ . During the latter half period of  $T$ ,  $T_1$  was off and  $T_2$  was on, so that the  $C_2$  was charged by  $C_1V_I$ . The output voltage of this circuit in a period was

$$V_o(nt) = V_o((n-1)t) - \frac{C_1}{C_2}V_I((n-1)t). \quad (8)$$

The tent map circuit consisted of the above circuit as shown in Figure 3. Parts I and III implemented the function of  $y = 2x$ , and parts II and III implemented the function of  $y = 2(1-x)$ . Part IV was a circuit for holding and delay.

Figure 4 shows the control logic for A/D conversion. First, start signal got high, and switch  $J_0$  connected the input signal with the part I and part II. After delay of  $t_1$ ,  $D$  trigger produces a switch instruction; if  $0 \leq V_i \leq 0.5$ , then  $J_1$  was switched on; if  $0.5 < V_i < 1$ , then  $J_2$  was switched on. At time  $t_1 + t_2$ , the control signal  $e$  switches were high, so that the charge of  $C_1$  or  $C_2$  was transferred to  $C_3$ , due to  $C_3 = (1/2)C_1 = (1/2)C_2$ , so that the input voltage was amplified doubly. In the meantime,  $C_4$  was also charged. At the next time 0, the  $o$  switches were switched off and  $e$  switches were switched on, so that the charge of  $C_4$  was transferred to  $C_5$  and results in  $V_{C_5} = V_{C_4}$ . At the same time,  $J_0$  was disconnected with input signal  $V_0$ , so that an iterative feedback loop exists in the A/D conversion, and  $C_1, C_2$  were charged by the same feedback voltage to realize  $y = 2x$  and  $y = 2(1-x)$ . This circle continuously iterated  $N$  times.

Hence the  $N$  binary bits output of  $D$  trigger was the Gray-code sequence of A/D conversion

$$g_k = \bar{Q}_k \quad (k = 1, 2, 3, \dots), \quad (9)$$

$$b_0 = Q_0.$$

The above  $g_k$  sequence, the initial condition  $b_0 = Q_0$  into (6), and Bernoulli binary sequence  $b_k$  ( $k = 0, 1, 2, \dots$ ) could be obtained.

#### 5. Array Tactile Sensor Signal Acquisition System Based on Tent Map of A/D Circuit

The schematic diagram of signal amplification of array tactile sensor analog and A/D conversion based on tent map circuit is shown in Figure 5. According to the arrangements of the timing control circuit, line scan circuit sent the periodic excitation signal to  $m$ -line array sensitive element, while the read circuit read the output signal of  $n$  column in parallel. The  $n$ -column signal was generated by simultaneously signaling nonlinear amplification and analog-digital conversion of  $n$  of A/D converters based on tent map. The resulting Gray-code sequence was sent directly to the computer which will

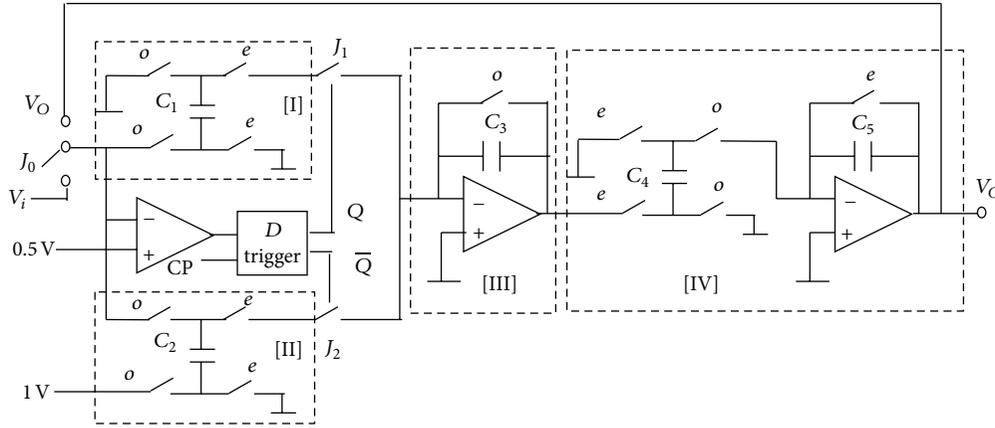


FIGURE 3: The circuit of tent map for A/D conversion.

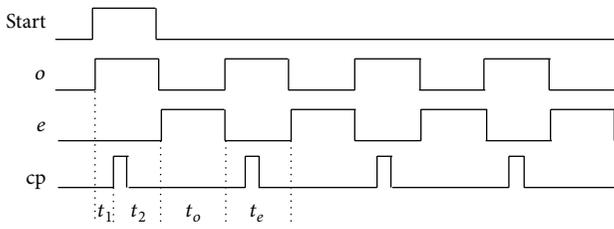


FIGURE 4: Circuit logic for the A/D conversion.

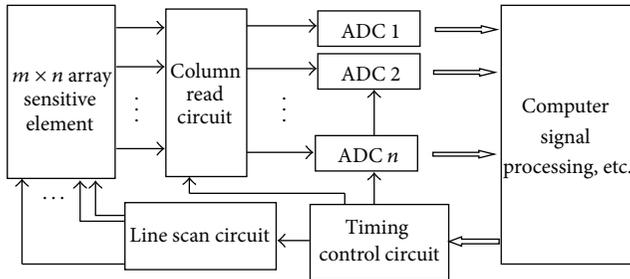


FIGURE 5: The schematic diagram of array tactile sensor signal acquisition system based on tent map of A/D circuit.

complete the conversion of Gray code to binary code. Then, under the control of timing logic, the read  $n$ -column signal on the next line, and the completed amplification and A/D conversion, after obtaining an  $m \times n$  tactile image signal by the computer, we could process tactile signals.

We could carry out the amplification and A/D conversion experiments sensitive element signal of  $16 \times 16$  microarray tactile sensor based on the above circuit. The results of the A/D conversion for eight sensitive elements in the 8th line were shown in Table 1, which indicates that A/D circuits based on the tent map could effectively achieve the amplification and A/D conversion of a small signal.

Figure 6 is the output of a  $16 \times 16$  micro array tactile sensor manufactured by us based on the tent map circuit when a very light hexagon aluminum flake was put on it. The measurement range of each tactile sensing unit was from 0.01 N to 10 N, which shows that the proposed tent map

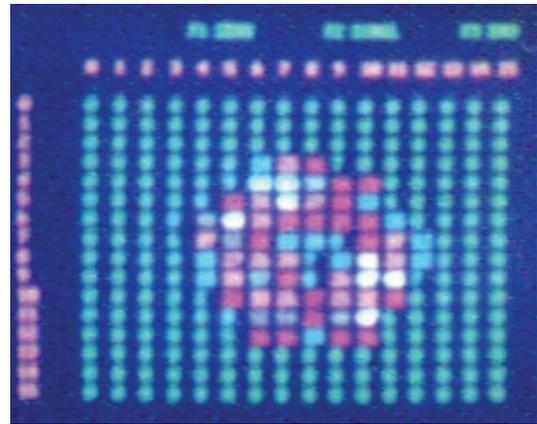


FIGURE 6: The output of a  $16 \times 16$  array tactile sensor based on the tent map circuit when a very light hexagon aluminum flake was put on it.

based A/D conversion circuits had the advantages of large amplification range and high resolution. By comparison with the conventional high precision array tactile sensor [17], our method is more cost effective and easier to realize.

## 6. Conclusion

This paper presents a novel A/D conversion circuit for robot array tactile sensor of unique properties. The circuit makes use of unique advantage for tent map sensitive to small signal circuit and nonlinear transform and has conditioning amplification and A/D conversion function integration advantages, simple circuit, and easy integration to realize. This method can achieve the parallel sampling of multichannel tactile signal and A/D converter, which can meet the real-time requirements of signal acquisition for the large scale array tactile sensor. This experiment gives the effectiveness of this method.

## Acknowledgments

This work was supported by the Science and Technology Project of National Energy Administration (no. NY20110702-1)

and the Technology Project of State Grid Corporation of China in 2013-2014 (project name is Study of Testing and Detection Technology for Electric Vehicle Charging and Battery Swap Infrastructure).

[17] H.-K. Kim, S. Lee, and K.-S. Yun, "Capacitive tactile sensor array for touch screen application," *Sensors and Actuators A*, vol. 165, no. 1, pp. 2–7, 2011.

## References

- [1] H. Yousef, M. Boukallel, and K. Althoefer, "Tactile sensing for dexterous in-hand manipulation in robotics—a review," *Sensors and Actuators A*, vol. 167, no. 2, pp. 171–187, 2011.
- [2] J. Ma and A. Song, "Fast estimation of strains for cross-beams six-axis force/torque sensors by mechanical modeling," *Sensors*, vol. 13, no. 5, pp. 6669–6686, 2013.
- [3] A. Song, J. Wu, G. Qin, and W. Huang, "A novel self-decoupled four degree-of-freedom wrist force/torque sensor," *Measurement*, vol. 40, no. 9-10, pp. 883–891, 2007.
- [4] F. Beyeler, S. Muntwyler, and B. J. Nelson, "A six-axis MEMS force-torque sensor with micro-Newton and nano-Newton-meter resolution," *Journal of Microelectromechanical Systems*, vol. 18, no. 2, pp. 433–441, 2009.
- [5] J. Ma, A. Song, and J. Xiao, "A robust static decoupling algorithm for 3-axis force sensors based on coupling error model and  $\epsilon$ -SVR," *Sensors*, vol. 12, no. 11, pp. 14537–14555, 2012.
- [6] J.-H. Kim, J.-I. Lee, H.-J. Lee, Y.-K. Park, M.-S. Kim, and D.-I. Kang, "Design of flexible tactile sensor based on three-component force and its," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2578–2581, April 2005.
- [7] M. H. Lee and H. R. Nicholls, "Tactile sensing for mechatronics—a state of the art survey," *Mechatronics*, vol. 9, no. 1, pp. 1–31, 1999.
- [8] A. Song, Y. Han, H. Hu, L. Tian, and J. Wu, "Active perception-based haptic texture sensor," *Sensors and Materials*, vol. 25, no. 1, pp. 1–15, 2013.
- [9] J.-H. Lee and C.-H. Won, "High-resolution tactile imaging sensor using total internal reflection and nonrigid pattern matching algorithm," *IEEE Sensors Journal*, vol. 11, no. 9, pp. 2084–2093, 2011.
- [10] J. Dargahi and S. Najarian, "Advances in tactile sensors design/manufacturing and its impact on robotics applications—a review," *Industrial Robot*, vol. 32, no. 3, pp. 268–281, 2005.
- [11] S. Saga, H. Kajimoto, and S. Tachi, "High-resolution tactile sensor using the deformation of a reflection image," *Sensor Review*, vol. 27, no. 1, pp. 35–42, 2007.
- [12] N. Singh and A. Sinha, "Chaos-based secure communication system using logistic map," *Optics and Lasers in Engineering*, vol. 48, no. 3, pp. 398–404, 2010.
- [13] L. Zhang, A. Song, and J. He, "Stochastic resonance of a subdiffusive bistable system driven by Lévy noise based on the subordination process," *Journal of Physics A*, vol. 42, no. 47, Article ID 475003, 2009.
- [14] A. Song, J. Duan, J. Wu, and H. Li, "Design 2D nonlinear system for information storage," *Chaos, Solitons and Fractals*, vol. 41, no. 1, pp. 157–163, 2009.
- [15] D. Rousseau, J. R. Varela, and F. Chapeau-Blondeau, "Stochastic resonance for nonlinear sensors with saturation," *Physical Review E*, vol. 67, no. 2, Article ID 021102, 6 pages, 2003.
- [16] X. Yi, "Hash function based on chaotic tent maps," *IEEE Transactions on Circuits and Systems II*, vol. 52, no. 6, pp. 354–357, 2005.

## Research Article

# Rapid 3D Modeling and Parts Recognition on Automotive Vehicles Using a Network of RGB-D Sensors for Robot Guidance

**Alberto Chávez-Aragón, Rizwan Macknojjia, Pierre Payeur, and Robert Laganière**

*Faculty of Engineering, School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada K1N 6N5*

Correspondence should be addressed to Alberto Chávez-Aragón; [achavez@uottawa.ca](mailto:achavez@uottawa.ca)

Received 15 March 2013; Revised 9 July 2013; Accepted 12 July 2013

Academic Editor: Lei Wang

Copyright © 2013 Alberto Chávez-Aragón et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents an approach for the automatic detection and fast 3D profiling of lateral body panels of vehicles. The work introduces a method to integrate raw streams from depth sensors in the task of 3D profiling and reconstruction and a methodology for the extrinsic calibration of a network of Kinect sensors. This sensing framework is intended for rapidly providing a robot with enough spatial information to interact with automobile panels using various tools. When a vehicle is positioned inside the defined scanning area, a collection of reference parts on the bodywork are automatically recognized from a mosaic of color images collected by a network of Kinect sensors distributed around the vehicle and a global frame of reference is set up. Sections of the depth information on one side of the vehicle are then collected, aligned, and merged into a global RGB-D model. Finally, a 3D triangular mesh modelling the body panels of the vehicle is automatically built. The approach has applications in the intelligent transportation industry, automated vehicle inspection, quality control, automatic car wash systems, automotive production lines, and scan alignment and interpretation.

## 1. Introduction

Robot manipulation and navigation require efficient methods for representing and interpreting the surrounding environment. Industrial robots, which work in controlled environments, are typically designed to perform only repetitive and preprogrammed tasks. However, robots working in dynamic environments demand reliable methods to interpret their surroundings and are submitted to severe time constraints. Most existing solutions for robotic environment representation and interpretation make use of high-cost 3D profiling cameras, scanners, sonars, or combinations of them, which often result in lengthy acquisition and slow processing of massive amounts of information. The extreme acquisition speed of the Kinect's technology meets requirements for rapidly acquiring models over large volumes, such as that of automotive vehicles. The performance, affordability, and the growing adoption of the Kinect for robotic applications supported the selection of the sensor to develop the robotic inspection station operating under multisensory visual guidance. The method presented in this work uses a set of Kinect depth sensors properly calibrated to collect visual information as well as

3D points from different regions over vehicle bodyworks. A dedicated calibration methodology is presented to achieve accurate alignment between the respective point clouds and textured images acquired by Kinect sensors distributed in a collaborative network of imagers to provide coverage over large surfaces. First, the sensing system uses computer vision and machine learning techniques for determining the location and category of a vehicle and some areas of interest over the bodywork. Then, the 3D readings are aligned using the extrinsic parameters between the Kinect units. Finally, a 3D triangle mesh, modeling the lateral panels of the vehicle, is built and serves as input to guide a manipulator robot that will interact with the surface. The experiments reported in this work are the result of processing images and point clouds of side panels of automobiles. Nevertheless, the method can be adapted easily to recognize other types of objects.

This work contributes to the robotic vision field by proposing a simple and efficient methodology for automatic 3D surface modeling of large vehicle parts via the coordinated and integrated operation of several RGB-D sensor heads; a dedicated methodology for extrinsic calibration of Kinect sensors, as well as a rapid algorithm for triangle meshing

which takes advantage of the structure of the point clouds provided by the Kinect sensors.

## 2. Related Work

In the recent years the Kinect device has been widely adopted as an indoor sensor for robotics and human-computer interaction applications. The sensor is a multiview structured lighting system, containing an RGB camera, an infrared (IR) camera, and an infrared laser projector equipped with a microgrid that artificially creates a predefined IR pattern over the imaged surface. The sensor is capable of collecting depth information for each pixel in a color image, which opens the door to a great variety of applications. Lately, two dominant streams of research have been pursued with Kinect technology: (1) the investigation of the technology behind the device, analysis of its properties, performance, and comparison with other depth sensors; (2) the development of applications of the Kinect technology in fields such as robotics, user interfaces, and medicine among others. The present work addresses both categories.

Among the numerous examples of applications for the Kinect technology that rapidly appeared in the literature, Zhou et al. [1] proposed a system capable of scanning human bodies using multiple Kinect sensors arranged in a circular ring. Maimone and Fuchs [2] presented a real-time telepresence system with head tracking capabilities based on a set of Kinect units. They also contributed an algorithm for merging data and automatic color adjustment between multiple depth data sources. An application of Kinect in the medical field for position tracking in CT scans was proposed by Noonan et al. [3]. They tracked the head of a phantom by registering Kinect depth data to high resolution CT template of a head phantom. Rakprayoon et al. [4] used a Kinect sensor for obstacle detection of a robotic manipulator. In [5], Berger et al. originally used multiple Kinect sensors for aerodynamic studies of 3D objects. They captured and visualized gas flow around objects with different properties. Smisek et al. [6] and Park et al. [7] conducted analyses regarding Kinect's depth resolution, accuracy with stereo resolution reconstruction, and camera calibration as well as a comparison with a laser scanner. For simultaneous calibration of the Kinect sensor, different approaches have been proposed. Burrus [8] proposed to use traditional techniques for calibrating the Kinect color camera and manual selection of the corners of a checkerboard for calibrating the depth sensor. Gaffney [9] described a technique to calibrate the depth sensor by using 3D printouts of cuboids to generate different levels in depth images. The latter, however, requires an elaborate process to construct the target. Berger et al. [10] used a checkerboard where black boxes were replaced with mirroring aluminium foil therefore avoiding the necessity of blocking the projector when calibrating the depth camera.

With regard to the depth data of the Kinect sensor, it is known that it suffers from quantization noise [6, 11] that increases as the distance to the object increases. The resolution also decreases with the distance [11]. The depth map may also contain occluded and missing depth areas mainly due

to the physical separation between the IR projector and the IR camera and to the inability to collect sufficient IR signal reflection over some types of surfaces. These missing values can however be approximated by filtering or interpolation [2, 12].

Concerning the automated detection of vehicle parts, a variety of computer vision systems have been developed in the past that aimed at detecting regions of interest in images of vehicles. Among popular applications in this field, the inspection of products on assembly lines stands out. Some of these systems used methods to simultaneously locate many reference points or many regions of interest [13, 14]. To manage the semantic information in the problem domain, Kiryakov et al. [15] used templates and similarity measures to evaluate the correct position of a template over an image. For the visual detection of features of interest in images some authors have reported the successful use of a technique proposed by Viola and Jones called cascade of boosted classifiers (CBC) [16]. This technique has proven to be useful in detecting faces, wheels, back views of cars, and license plates among others [17, 18]. While applications of previous research works are mainly in the area of intelligent transportation systems (ITS), these concepts can advantageously be transposed for applications in robotic guidance.

## 3. Proposed RGB-D Acquisition Framework

The work presented here aims at the integration of information from multiple RGB-D sensors to achieve fully automated and rapid 3D profiling of bodywork regions over automotive vehicles. The approach estimates the shape over selected regions to be reconstructed based on the detection of features of interest on vehicle body panels. Once the location of the regions of interest is known, the approach reconstructs the panels' shape using information provided by a set of Kinect sensors placed conveniently which collect visual and 3D information from the vehicle.

The final goal of the system being developed is to support the real-time navigation of a robotic arm in proximity of the vehicle in order to perform a series of tasks (e.g., cleaning, maintenance, inspection) while it is interacting with the vehicle surface. The work reported in this paper focuses mainly on the robotic vision stage.

Figure 1 shows various environments used to develop and test the proposed system. In Figure 1(a), an indoor laboratory environment is depicted where an actual piece of automotive bodywork was used for early development along with a mockup car door model. In Figure 1(b) a multilevel semioutdoor parking garage is shown where real full-size vehicles were imaged to validate the accuracy of calibration over the network of RGB-D sensors. The parking garage infrastructure prevented the direct sunlight from causing interference with the IR image components of the Kinect units. Natural light coming from windows and open walls as well as electric lamps lighted up the scene. The experiments demonstrated that these sources of light did not trouble the sensors' technology.

The layout of a vehicle scanning station is shown in Figure 2. Yellow lines delimit the area where the vehicle stops,



FIGURE 1: Indoor and semioutdoor environments used to develop and test the proposed acquisition stage.

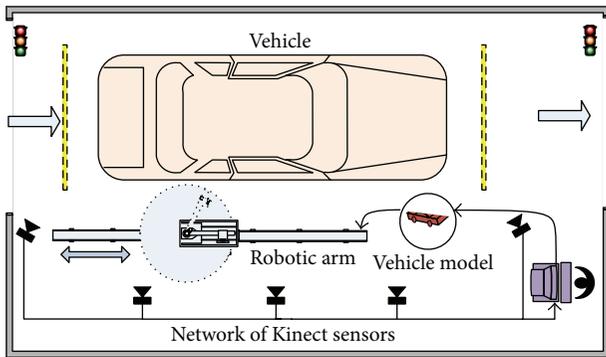


FIGURE 2: System layout of the proposed scanning system: yellow lines delimit the area where the vehicle stops, while depth and color information is collected.

while depth and color information is collected. At the beginning, the robotic arm which can be moved on rails is positioned at the end of the track within the blind spots of the sensors. Then, the vehicle enters the scanning area and stops in the designated space. The Kinect sensors collect color and depth information over the entire length of the vehicle within 10 seconds. The information is then processed in order to construct a 3D model of the bodywork panels on the vehicle. The whole scanning and modeling process is meant to be fast, in order to support high inspection cadence. The latter criterion was the main factor to support adoption of the Kinect technology in this application in spite of its limited depth resolution and sensitivity to ambient lighting conditions.

Overall the scanning and 3D textured modeling processes are completed in 30 seconds, which rapidly makes the information available for robot guidance. The data processing pipeline of the sensory information provided by the Kinect devices is shown in Figure 3. Rounded rectangles represent the different modules that are part of the process. Those modules within the dotted rectangle operate on-line, while the calibration procedure out of the dotted rectangle is executed off-line and only once prior to inspection operation. Rectangles represent the inputs and output of the system.

**3.1. System Configuration.** The scanning system for collecting color and depth information over the whole lateral view of a vehicle bodywork consists in three Kinects positioned in a single row (sensor's baseline) with their viewing axes perpendicular to the bodywork and two extra sensors, which cover

TABLE 1: Parameters of the proposed scanning system.

	IR camera	RGB camera
Horizontal field of view	57°	63°
Vertical field of view	45°	50°
Distance between cameras	1.3 m	1.3 m
Height of the sensors above the ground	1 m	1 m
Distance between baseline cameras and vehicle	2 m	2 m
Horizontal overlapping area between two cameras	0.85 m	1.15 m
Coverage area for each sensor	4.7 m × 1.65 m	5 m × 1.85 m
Total coverage area for the baseline depth sensors	6 m × 1.65 m	6.3 m × 1.85 m

partially the front and back areas, rotated toward the vehicle in such a way that their viewing axes form a 65 degree angle with respect to the sensor's baseline.

This configuration can be replicated on the other side of the vehicle for a complete 360 degree view. As detailed in Table 1, the sensors are positioned at 1 m above the ground and parallel to the floor. The cameras were set up about 2 m away from the area where the vehicle is positioned. This configuration permits to meet the following requirements to cover the entire side of a car: (1) a minimum coverage area of  $4.15 \times 1.5$  m, which is the typical size of a sedan vehicle; (2) collection of depth readings in the range of 0.8 to 3 m, which is the range where Kinect performs well; (3) an overlapping area in the range of 0.5 m to 1 m, between contiguous sensors to ensure accurate external calibration process and point cloud alignment. Figure 4 depicts the acquisition system. It is worth mentioning that the proposed acquisition system can be easily adapted for larger vehicles by including extra sensors in the sensor's baseline.

**3.2. Interference Issues.** Within the overlapping regions between two contiguous Kinect sensors, interference might happen between the sensors since all Kinect devices project a pattern of infrared points at the same wavelength to create their respective depth map. This produces small holes on the depth maps of overlapping sensors. To prevent this problem, the data is collected sequentially over different time slots. In the first time slot, sensors  $K_0$  and  $K_2$  simultaneously collect

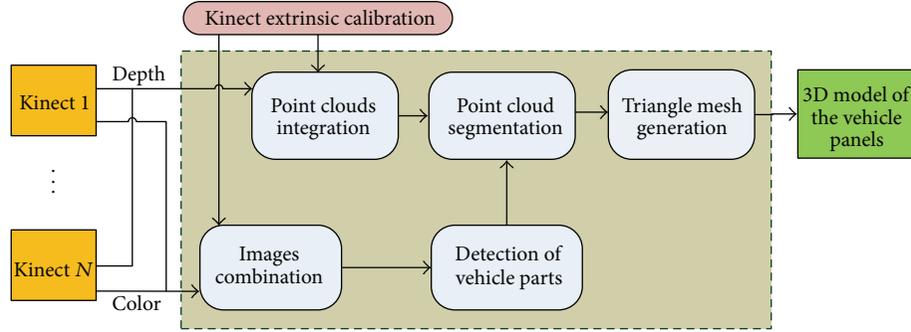


FIGURE 3: Main components of the RGB-D scanning and modeling approach.

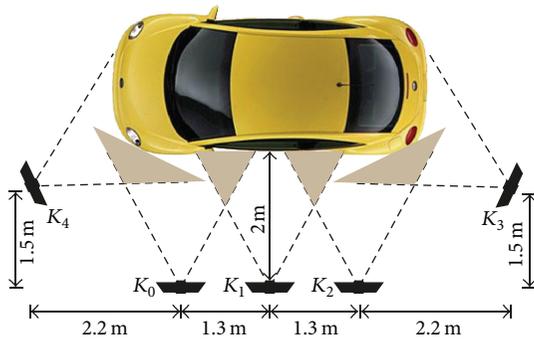


FIGURE 4: Configuration of the acquisition system. Five Kinect sensors are used to collect color and depth information over the entire set of lateral panels of a vehicle.

their respective information (see Figure 4). Then, sensors  $K_1$ ,  $K_3$ , and  $K_4$  scan the central, back, and front regions of the vehicle. The delay between the shots is the time needed to shut down the first group of devices and initialize the next one. This process is performed by the Kinect drivers and takes few seconds. The OpenNI framework was used to control the Kinect for Xbox 360 version of the sensors; this framework does not provide means to shut down only the infrared projector.

#### 4. Calibration of a Network of Kinect Sensors

Kinect technology consists of a multiview system that provides three outputs: an RGB image, an infrared image, and a depth image for each sensor. Arranging a group of sensors as a collaborative network of imagers permits to enlarge the overall field of view and to model large objects, such as automotive vehicles. For an adequate operation of the network a precise mapping between color and infrared must be achieved. For this purpose an internal calibration procedure that estimates the intrinsic parameters of each camera within every device as well as the extrinsic parameters between the RGB and the IR camera inside a given Kinect is required, along with an external calibration process that provides accurate estimates of the extrinsic parameters in between the respective pairs of Kinect devices. A procedure dedicated to Kinect sensors is proposed and detailed later.

#### 4.1. Internal Calibration

**4.1.1. Intrinsic Parameters Estimation for Built-In Kinect Cameras.** The internal calibration procedure includes the estimation of the respective intrinsic parameters for the color and the IR sensors, which are the focal length  $(f_x, f_y)$ , the principal point  $(O_x, O_y)$ , and the lens distortion coefficients  $(k_1, k_2, p_1, p_2, k_3)$  [19]. Because the RGB and IR cameras exhibit different color responses, the proposed calibration technique uses a regular chessboard target of size  $9 \times 7$  that is visible in both sensors' spectra. During internal calibration the Kinect's IR projector is blocked by overlapping a mask on the projector window. The IR projector otherwise introduces noise over the IR image as shown in Figure 5(a), and without projection, the image is too dark as shown in Figure 5(b). Therefore standard external incandescent lamps are added to illuminate the checkerboard target, Figure 5(c). The color image is not affected by the IR projection and creates a clear pattern, Figure 5(d).

The checkerboard was printed on a regular A3 size paper, which does not reflect back the bright blobs due to the external incandescent lamps in the IR image plane. To ensure the best calibration results, 100 images were collected from both the color and the IR cameras. Both images were synchronized in each frame, so that they could be used for extrinsic calibration between the cameras. To estimate the intrinsic parameters, each Kinect is calibrated individually using Zhang's camera calibration method [19]. The method is applied 10 times on 30 images randomly selected among the 100 captured images. The reprojection error is also calculated for each iteration, which is a measure of the deviation of the camera response to the ideal pinhole camera model. The reprojection error is calculated as the RMS error of all the target calibration points.

After calibration, both the RGB and IR cameras achieve reprojection error between 0.12 and 0.16 pixels, which is better than the original performance given by the default manufacturer calibration of the Kinect sensor. The reprojection error without calibration of the IR camera is greater than 0.3 pixel and that of the color camera is greater than 0.5 pixel. The focal length of the IR camera is larger than that of the color camera, that is, the color camera has a larger field of view. It is also apparent that every Kinect sensor has slightly different intrinsic parameters. This confirms the need for a formal

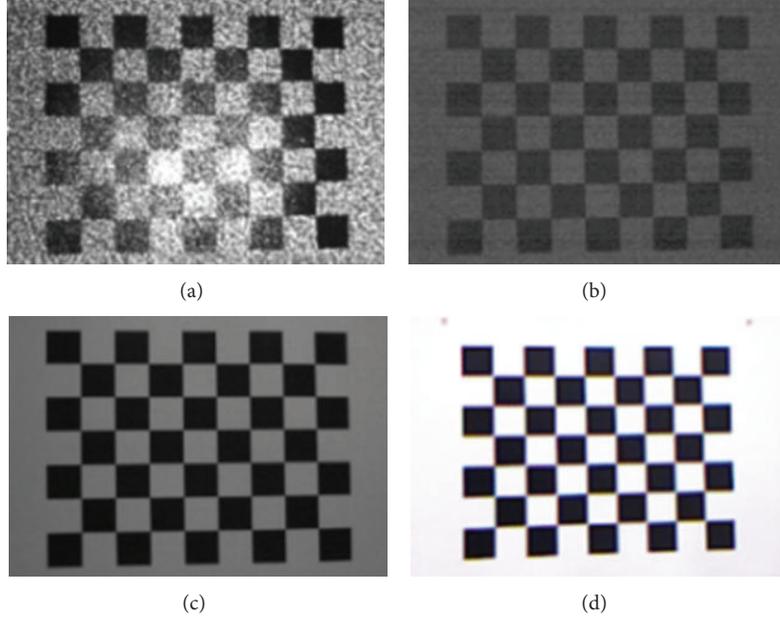


FIGURE 5: Views of the checkerboard in different configurations: (a) IR image with IR projector, (b) IR image without IR projector, (c) IR image with incandescent lighting and without projector, and (d) color image.

intrinsic calibration to be performed on each device to support accurate data registration.

**4.1.2. Extrinsic Parameters Estimation for Built-In Kinect Cameras.** The respective location of the color and IR cameras within each Kinect unit is determined by stereo calibration. The camera calibration method proposed by Zhang [19] also provides the location of the checkerboard target with respect to the camera coordinate system. If the target remains fixed for both cameras, then the position between both cameras is defined by (1)

$$H = H_{\text{RGB}}H_{\text{IR}}^{-1}, \quad (1)$$

where  $H$  is the homogenous transformation matrix (consists of  $3 \times 3$  rotation matrix  $R$  and  $3 \times 1$  translation vector  $T$ ) from the RGB camera to the IR camera,  $H_{\text{IR}}$  is the homogenous transformation matrix from the IR camera to the checkerboard target, and  $H_{\text{RGB}}$  is the homogenous transformation from the RGB camera to the checkerboard target. The translation and rotation parameters between the RGB and IR sensors are shown in Table 2 for five Kinect sensors. The internal extrinsic calibration parameters allow to accurately relate the color to depth data collected by a given Kinect device.

**4.1.3. Registration of Color and Depth Information within a Given Kinect Device.** The Kinect sensor does not provide the registered color and depth images. Once the internal intrinsic and extrinsic parameters are determined for a given Kinect device, the procedure to merge the color and depth based on the estimated registration parameters is performed as follows. The first step is to properly relate the IR image and the depth

TABLE 2: Internal extrinsic calibration of embedded sensors.

Translation (cm) and rotation (degree) between RGB and IR						
Sensor	$T_x$	$T_y$	$T_z$	$R_x$	$R_y$	$R_z$
$K_0$	2.50	0.0231	0.3423	0.0017	0.0018	-0.0082
$K_1$	2.46	-0.0168	-0.1426	0.0049	0.0032	0.0112
$K_2$	2.41	-0.0426	-0.3729	0.0027	0.0065	-0.0075
$K_3$	2.49	0.0153	0.2572	-0.0046	0.0074	0.0035
$K_4$	2.47	0.0374	0.3120	0.0052	0.0035	0.0045

image. The depth image is generated from the IR image but there is a small offset between the two, which is introduced as a result of the correlation performed internally during depth calculation. The offset is 5 pixels in the horizontal direction and 4 pixels in the vertical direction [6]. After removing this offset using (2), each pixel of the depth image exactly maps the depth of the corresponding pixel in the IR image. Therefore, the calibration parameters of the IR camera can be applied on the depth image considered

$$dp(x, y) = ds(x - 5, y - 4), \quad (2)$$

where  $x$  and  $y$  are the pixel location,  $ds(x, y)$  is the offsetted depth value affecting the Kinect depth sensor, and  $dp(x, y)$  is the corrected depth value. The second step consists in transforming both the color and the depth images to compensate for radial and tangential lens distortion using OpenCV undistort function [20]. This function estimates the geometric transformation on the images using the distortion parameters and provides the undistorted color image and depth image ( $du(x, y)$ ). The next step is to determine the 3D coordinates

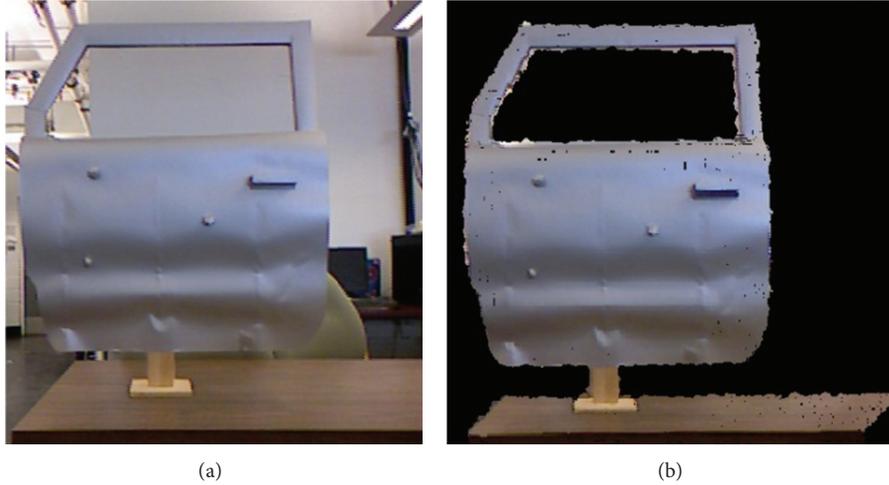


FIGURE 6: Accurate integration of color and depth images.

corresponding to each point in the undistorted depth image, using (3)

$$\begin{aligned} X_{\text{IR}} &= \frac{(x - O_{x,\text{IR}}) du(x, y)}{f_{x,\text{IR}}}, \\ Y_{\text{IR}} &= \frac{(y - O_{y,\text{IR}}) du(x, y)}{f_{y,\text{IR}}}, \\ Z_{\text{IR}} &= du(x, y), \end{aligned} \quad (3)$$

where  $(X_{\text{IR}}, Y_{\text{IR}}, Z_{\text{IR}})$  are the 3D point coordinates of pixel  $(x, y)$  in the depth image with respect to the IR camera reference frame,  $(x, y)$  is the pixel location,  $(f_{x,\text{IR}}, f_{y,\text{IR}})$  is the focal length of the IR camera,  $(O_{x,\text{IR}}, O_{y,\text{IR}})$  is the optical center of the IR camera, and  $du(x, y)$  is the depth of a pixel in the undistorted depth image. Next, the color is assigned from the RGB image to each 3D point  $P_{\text{IR}}(X_{\text{IR}}, Y_{\text{IR}}, Z_{\text{IR}})$ . The color is mapped by transforming the 3D point  $P_{\text{IR}}$  into the color camera reference frame using the internal extrinsic camera parameters and then reprojecting that point on the image plane of the RGB camera using the intrinsic parameters to find the pixel location of the color in the undistorted color image using (4)

$$\begin{aligned} P_{\text{RGB}}(X_{\text{RGB}}, Y_{\text{RGB}}, Z_{\text{RGB}}) &= R \cdot P_{\text{IR}} + T, \\ x &= \left( \frac{X_{\text{RGB}} f_{x,\text{RGB}}}{Z_{\text{RGB}}} \right) + O_{x,\text{RGB}}, \\ y &= \left( \frac{Y_{\text{RGB}} f_{y,\text{RGB}}}{Z_{\text{RGB}}} \right) + O_{y,\text{RGB}}, \end{aligned} \quad (4)$$

where  $P_{\text{RGB}}$  is the 3D point with respect to the color camera reference frame,  $R$  and  $T$  are the rotation and translation parameters from the color camera to the IR camera, and  $(x, y)$  is the pixel location of color information in the undistorted color image.

Figure 6(a) shows a mockup car door as imaged from the color camera; Figure 6(b) shows the colored depth information in the interval 0–2.5 m from the slightly different point of view of the IR camera, while keeping the Kinect sensor static with respect to the panel. The difference in position and orientation between the two cameras contained in the Kinect unit is accurately compensated by the estimated extrinsic parameters obtained from internal calibration.

**4.2. External Calibration of Kinect Sensors with a Best-Fit Plane Method.** The last set of parameters estimated in the calibration process is the extrinsic ones, which is the relative position and orientation between every pair of Kinect sensors. The external calibration is performed between pairs of IR cameras over the network of sensors because depth information is generated with respect to these cameras. The concept behind the proposed method, named here best-fit plane calibration method, is to determine, for every pair of sensors, the position and orientation of a fixed planar chessboard in real world coordinates. Knowing the orientation of the plane from two different points of view (i.e., two Kinect sensors), it is possible to estimate the relative orientation and position change between the sensors.

The procedure developed for external calibration involves positioning a standard planar chessboard target within the visible overlapping regions of any two Kinect sensors. Unlike most calibration techniques in the literature, in this method there is no need to move the checkerboard to image it from multiple views. On the contrary, a fixed target increases the performance of the method. The result is a rigid body transformation that best aligns the data collected by a pair of RGB-D sensors. Figure 7 depicts the proposed scanning system during the calibration process.

The proposed technique takes advantage of the rapid 3D measurement technology embedded in the sensor and provides registration accuracy within the range of the depth measurements resolution available with Kinect. An important advantage of this method is the fact that it is unnecessary to



FIGURE 7: Placement of calibration target during calibration of Kinects  $K_0$  and  $K_4$ .

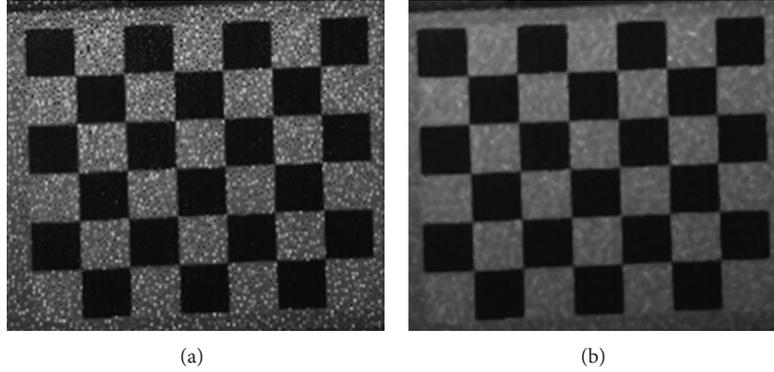


FIGURE 8: IR image of the checkerboard target for external calibration: (a) effect of the projected IR pattern, (b) filtered image using a median filter of size  $3 \times 3$ .

cover the Kinect infrared projector to perform this phase of the calibration, which facilitates manipulations when remotely dealing with the network of Kinect devices.

The method consists in finding a normal vector and the center of the checkerboard plane, which define the relative orientation and translation of the checkerboard plane. The first step is to compute the 3D coordinates of the corners on the checkerboard with respect to the IR camera frame, using (3). When the checkerboard target is positioned in front of a Kinect sensor, the IR projector pattern appears on the checkerboard target as shown in Figure 8(a). This pattern creates noise and makes it difficult to extract the exact corners. Since the noise is similar to salt and pepper noise, a median filter of size  $3 \times 3$  provides a good reduction in the noise level without blurring the image, as shown in Figure 8(b).

Moreover, the extracted points are not entirely mapped over a single plane because of quantization effects in the Kinect depth sensor. Therefore, the corner points are used to estimate the three-dimensional plane (5) that minimizes the orthogonal distance between that plane and the set of 3D points. The equation of the plane then permits to estimate the orientation in 3D space of the target with respect to the IR camera

$$z = Ax + By + C. \quad (5)$$

Let the 3D coordinates of the  $n$  corners of the checkerboard target be  $S_1(x_1, y_1, z_1), S_2(x_2, y_2, z_2), \dots, S_n(x_n, y_n, z_n)$ ; then the systems of equations for solving the plane equation

are  $Ax_1 + By_1 + C = z_1, Ax_2 + By_2 + C = z_2, \dots, Ax_n + By_n + C = z_n$ . These equations can be formulated into a matrix problem

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}. \quad (6)$$

This overdetermined system is solved for the values of  $A$ ,  $B$ , and  $C$  with an orthogonal distance regression approach [21], which provides the best fit plane on those points. All the 3D points,  $S_n$ , are projected on the best fit plane,  $P_n$ . These points serve to define the center and the normal vector of the plane. However, projected points,  $P_n$ , do not represent the exact corners of the checkerboard. Therefore, the center of the plane cannot be defined only by the intersection of two lines passing close to the center. Figure 9(a) shows the set of possible pairs of symmetric corner points that generate lines passing close to the center.

The closest point to all intersections between these lines is selected as a center point  $O$ . Two points  $P_i$  and  $P_j$  are selected on the plane to define vectors  $\overrightarrow{OP_i}$  and  $\overrightarrow{OP_j}$ . The normal to the plane is then defined by the cross product:

$$z = \frac{\overrightarrow{OP_i} \times \overrightarrow{OP_j}}{|\overrightarrow{OP_i} \times \overrightarrow{OP_j}|}. \quad (7)$$

This normal is the unit vector of the  $z$ -axis of the checkerboard frame with respect to the RGB-D sensor. The unit

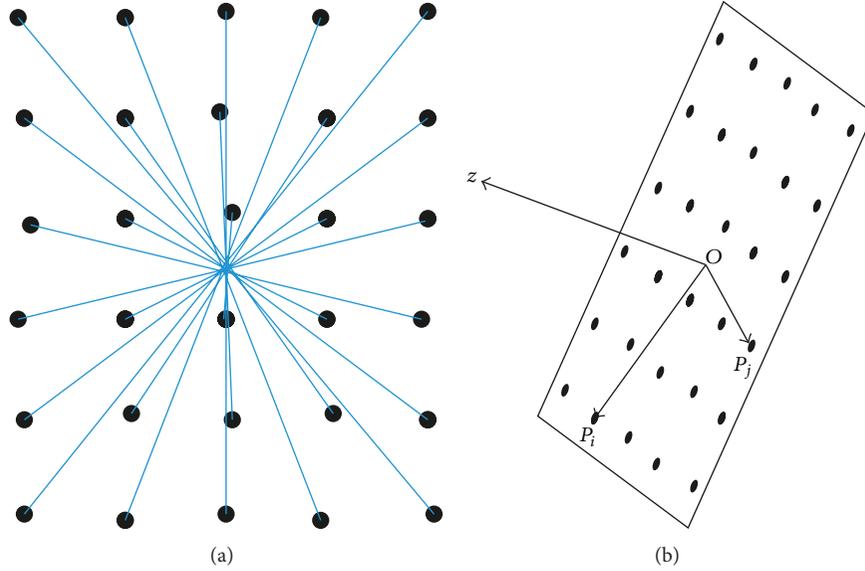


FIGURE 9: (a) Possible combination of lines passing through the center of the checkerboard, (b) the normal vector and the center of a checkerboard target.

vector of the  $y$ -axis of the checkerboard can be found by any two vertical points in the checkerboard. Let  $P_i$  and  $P_j$  be the two vertical points where  $P_i$  is the top end and  $P_j$  is the bottom end of a vertical line.  $N$  is the total number of possible combinations of vertical lines. The average unit directional vector can then be defined as

$$y = \frac{1}{N} \sum \frac{P_i - P_j}{|P_i - P_j|}. \quad (8)$$

This vector is the unit vector of the  $y$ -axis of the checkerboard frame with respect to the RGB-D sensor. The last unit vector for the  $x$ -axis can be found by a cross product, defined as

$$x = y \times z. \quad (9)$$

All the unit vectors of the coordinate frame of the checkerboard target can be combined to define the rotation matrix between the RGB-D sensor and the checkerboard frame as

$$R = \begin{bmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{bmatrix}. \quad (10)$$

The translation vector corresponds with the center of the checkerboard frame

$$T = [O_x \ O_y \ O_z]. \quad (11)$$

$R$  and  $T$  are the rotation and the translation matrices of the checkerboard frame with respect to the Kinect IR sensor. The position and orientation between two Kinect sensors can be determined by the following procedure. Let  $H_1$  and  $H_2$  be the homogenous transformations between Kinect 1 and the checkerboard and between Kinect 2 and the checkerboard, respectively, as shown in Figure 10. If the target remains

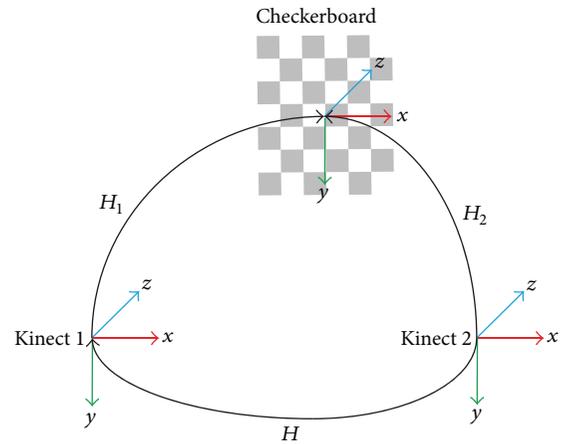


FIGURE 10: Extrinsic calibration of a pair of Kinect sensors.

fixed for both Kinect sensors, the geometrical transformation between the sensors is defined as follows:

$$H = H_2 H_1^{-1}, \quad (12)$$

where  $H$  is the homogenous transformation matrix from the Kinect 2 to the Kinect 1 sensor.

**4.3. Complete Calibration of All Cameras in the Network of Kinect Sensors.** The camera arrangement shown in Figure 4 includes overlapping regions between contiguous sensors marked in gray. During the calibration phase, the checkerboard target is successively placed within those areas for external calibration between every pair of neighbor Kinect IR sensors. Figure 7 shows the calibration target placed in the overlapping region between Kinect  $K_0$  and  $K_4$  during an experimental calibration procedure. External calibration is

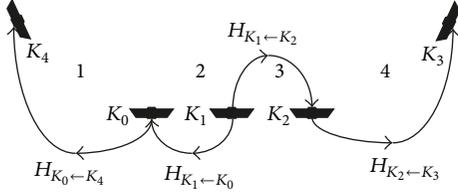


FIGURE 11: Calibration flow for the network of sensors.

performed in pairs using the best-fit plane calibration method detailed in Section 4.2. The center Kinect,  $K_1$ , is set as a base of reference for the setup. The relative calibration is then calculated between  $(K_1, K_0)$ ,  $(K_1, K_2)$ ,  $(K_2, K_3)$ , and  $(K_0, K_4)$ . Figure 11 shows the calibration flow for the network of Kinect sensors.

Kinects  $K_0$  and  $K_2$  have a direct relationship with  $K_1$ , immediately defined by the extrinsic calibration parameters obtained in each case, but  $K_4$  and  $K_3$  need to be related to  $K_1$  through an intermediate node, respectively,  $K_0$  and  $K_2$ . The relations between  $(K_1, K_4)$  and  $(K_1, K_3)$  are given by the following equations:

$$\begin{aligned} H_{K_1 \leftarrow K_4} &= H_{K_1 \leftarrow K_0} H_{K_0 \leftarrow K_4}, \\ H_{K_1 \leftarrow K_3} &= H_{K_1 \leftarrow K_2} H_{K_2 \leftarrow K_3}. \end{aligned} \quad (13)$$

## 5. Automatic Detection of Characteristic Areas over a Vehicle

Once the network of Kinect sensors distributed around one side of a vehicle is available and entirely calibrated, the proposed system determines the location of the vehicle in the scene and subsequently the location of a set of significant vehicle components. The purpose of recognizing specific areas of interest over a large object such as a vehicle is to speed up the modeling process and also to facilitate the guidance of the robot arm that will eventually interact with the vehicle to perform either inspection or maintenance tasks. Acquiring the knowledge about the location of dominant features over the vehicle reduces the amount of time spent on scanning at higher resolution to accurately drive the manipulator by focusing the operation only over selected areas. It also allows the robot to rapidly determine where to operate, as it is very unlikely that the robotic operation will be required over the entire vehicle for most typical inspection or maintenance tasks.

To achieve efficient and reliable detection and localization of characteristic areas over a vehicle, a visual detector of vehicle parts (VDVP) was previously introduced in [22] to operate on an image depicting a complete view of one side of a vehicle. The VDVP receives as an input a color image of a lateral view of the vehicle to determine the location of up to 14 vehicle parts. The method works with images of different types of vehicles such as 4-door sedan, 2-door sedan, 3-door hatchback, 5-door hatchback, SUV and pickup-trucks. Figure 12 illustrates the result of applying the method over a test image. Round areas indicate features detected by the classifiers; square regions mean that the locations of the features were

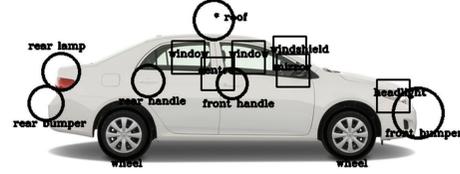


FIGURE 12: Automatic detection of parts of interest over a side view of a car.

inferred based on other known features. The VDVP method is revisited and integrated in this work.

The original method required at least a full lateral view of the vehicle which was provided by a standard color camera located far enough from the vehicle to image its entire length. To better integrate and to increase the compactness of the acquisition setup, determining the location of the car and the bodywork panels of the vehicle should be achieved by using the color information provided by the Kinect sensors. However, none of the Kinect sensors can collect a full view alone due to the limited horizontal field of view and depth of field of Kinect sensor technology. Therefore, it becomes necessary to merge the color images collected in a collaborative manner by sensors  $K_0, K_1, K_2$ . For that purpose, the assumption is made that the image planes of the group of Kinect sensors are parallel and they all are contained in a larger plane. Thus, in order to accurately merge the color information, only the translation vectors defining the relative position of the color cameras with respect to the central Kinect's color camera are required. These parameters were obtained during the calibration procedure described in Section 4 and can be computed by combining the internal and external extrinsic calibration parameters, as detailed in Section 4.3. The resulting composite image is used to determine the segments in each color image that correspond to both the vehicle and the parts of interest. Here the original VDVP method is extended such that the detection process for each part of interest is performed using individual images from each sensor, each representing a section of the vehicle rather than on the segmented panorama.

To determine the location of a car in the side view, the VDVP method relies on the successful detection of the car wheels [22]. To tackle this problem, the algorithm makes a decision based on the result of two techniques that estimate the position of the wheels: the Hough transform algorithm and a classifier based on Haar-like features trained for detecting wheels [16]. Then, the car is enclosed in a bounding box and segmented from the background. To set the dimension of the bounding box, a statistical analysis of the location of significant car features over a training set of images ( $t_s$ ) was conducted. The training set represents the 67% of a collection containing one hundred and ten color images. The remaining 33% of the images were used for evaluation purposes. Figure 13 shows the regions where features of interest are expected to be located. Each region of interest corresponds, from left to right, respectively, to *rear bumper*, *rear lamp*, *rear wheel*, *rear handle*, *rear window*, *roof*, *center*, *front handle*, *front window*, *windshield*, *lateral mirror*, *front wheel*, *head light*, and *front bumper*.



FIGURE 13: Regions containing the location of parts of interest for lateral views of vehicles, as a result of training over a representative collection of images of different types of automobiles.

Having recovered the position of the wheels in the scene, for parts of interest detection purposes, a polar coordinate system is established. The origin is the center of the rear wheel. Directions  $X$  and  $Y$  are at right angles to each other. The direction of the  $X$  axis matches with the directed line that goes from the origin (center of rear wheel) to the center of the front wheel.

Once the location of the wheels is known, the next step consists in determining the location of all other remaining features of interest,  $C_i$ , on the bodywork. Any feature of interest,  $C_i$ , characterizing a vehicle is defined in the polar coordinate system, as follows:

$$C_i = \{(r, \theta, \delta) : r, \delta \in R, 0 \leq \theta \leq \pi\}, \quad (14)$$

where  $r$  and  $\theta$  define the center of a feature of interest to be located, in polar coordinates, and  $\delta$  is the minimum radius of a circumference enclosing the region for a given part of interest. Let  $R_c$  be the region where a feature,  $C$ , is expected to be located.  $R_c$  is a 2-tuple  $(f_r, f_\theta)$  that can be represented by probability distribution functions over the polar map superimposed over the lateral image of the vehicle and defined as

$$\begin{aligned} f_r(r; \mu_r, \sigma_r^2) &= \frac{1}{\sigma_r \sqrt{2\pi}} e^{-((r-\mu_r)^2/2\sigma_r^2)}, \\ f_\theta(\theta; \mu_\theta, \sigma_\theta^2) &= \frac{1}{\sigma_\theta \sqrt{2\pi}} e^{-((\theta-\mu_\theta)^2/2\sigma_\theta^2)}. \end{aligned} \quad (15)$$

The radial and angular standard deviations ( $\sigma_r$  and  $\sigma_\theta$ ) and expectations ( $\mu_r$  and  $\mu_\theta$ ) in (15) are calculated experimentally from the training set,  $t_s$ . Consequently, these PDF functions define a probable searching area for each part of interest considered.

To achieve rotation, scale and translation invariance in the definition of the search zones, the direction of the vector pointing toward the center of the front wheel is used as the  $X$  axis, all the vectors pointing toward features of interest were normalized with respect to the length between the wheels' centers, and the origin of the coordinate system corresponds to the position of the rear wheel's center as it is shown in Figure 14(b). Up to this point a set of regions for each part of interest and for each image, in the segmented panorama, was defined. Next, for each car feature,  $C_i$ , to be detected, a search area,  $R_{c_i}$ , is defined. Then, a set of classifiers trained for detecting each feature of interest,  $C_i$ , is used. The detection method is constructed as a cascade of boosted classifiers based on Haar-like features. Classifiers were trained with the set of

images,  $t_s$ . Each detection performed by a classifier is ranked using the corresponding PDF functions ( $f_r, f_\theta$ ). False detections are discarded rapidly using this method as well.

**5.1. Determining the Position of Missing Vehicle Parts.** Vehicle panels are located using vehicle features on the bodywork. Figure 15 shows the classification rate for each car part. The classifiers were evaluated using the testing set that, as mentioned previously, contains the 33% of a collection of color images of vehicles. Twelve features are detected by the learning algorithms; the locations of the remaining features (windows) as well as the location of those characteristics that were not successfully detected by the classifiers are inferred as follows.

Let  $M$  be a geometrical model for a vehicle defined as follows:  $M = \{C_1, C_2 \dots C_n\}$ . That is,  $M$  defines the spatial relationship among the vehicle features of interest in an image by containing the positions of each car part.

Let  $M_c = \{C_1, C_2 \dots C_{n-k}\}$  a geometrical model constructed using the vehicle parts of interest successfully detected with the method proposed in the previous section,  $k$  being the number of missing features. Let  $G$  be the set of geometrical models for each image in the training set,  $t_s$ . A similarity function,  $L$ , which measures how adequate  $M_c$  and  $M$  are, can be defined as follows:

$$L(M_c, M) = \sum_{i=1}^k S(c_i) \cdot F(c_i), \quad (16)$$

where  $S(c_i)$  is the probability of the successful detection of the classifier for a particular feature,  $c_i$ . This probabilistic distribution was determined experimentally for each car part to be detected and is reported for dominant parts of interest in [22].  $F(c_i)$  defines the probability that a detected feature,  $c_i$ , was found in the right place considering the model,  $M$ , and the probability distribution associated with  $c_i$ . Therefore, for each feature,  $c_i$ , in  $M_c$ , the similarity function,  $L$ , is calculated using models in  $G$ . This way the most likely position of  $c_i$  is determined by choosing the lowest value for  $L$ . The partial model,  $M_c$ , is upgraded with the new location of  $c_i$ . This process is repeated until the full set of features is known. Finally, the model,  $M_c$ , not only provides accurate information about the location of the parts of interest but information about the type of car, size, location, and orientation of the vehicle since each model  $M$  in  $G$  beyond containing the spatial relationship among the vehicle features of interest also contains semantic information about the type of vehicle. Figure 16 shows some results obtained after applying the proposed technique for part detection in color images collected from three Kinect sensors over different lateral sections of a vehicle.

## 6. 3D Reconstruction of Vehicle Lateral Panels

The point clouds collected by the set of Kinect sensors are first aligned using the external calibration parameters previously calculated. Then, a segmentation procedure is applied to separate points contained within discovered regions of interest, using the VDVP method of Section 5, from the whole point cloud. For segmentation, the locations of the detected parts of

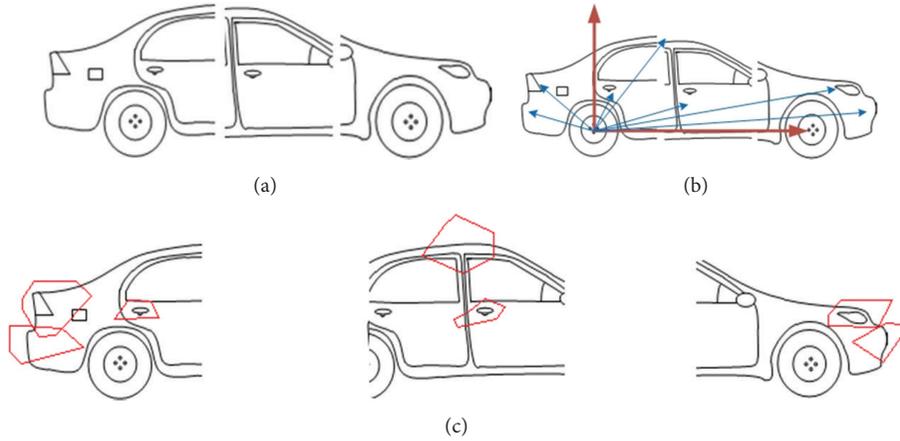


FIGURE 14: (a) Unrefined segmented panorama, (b) polar coordinate system centered on the rear wheel and the centers of each search zone, and (c) search zones  $R_c$  for each image.

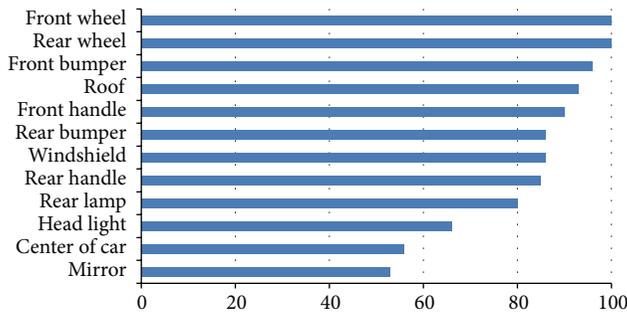


FIGURE 15: Classification rate according to car parts in percentage.

interest in color images are mapped into the 3D point cloud. The coordinates of the rear bumper, front bumper, roof, and wheels provide the width and height of a 3D bounding box enclosing the desired points. The bounding box depth is defined by adding and subtracting a fixed parameter (30 cm) to the average depth of the parts. This parameter was established experimentally, as folds and pleats of the surface of automotive vehicles usually do not go beyond this threshold. Next, a triangle mesh is built over the respective groups of points that correspond to each region of interest detected over the surface. This process results in a group of local 3D colored models that represent the shape and visual appearance of the surface of the vehicle in the specific regions of interest. A mesh is a topological structure typically constructed from unorganized points. The Delaunay triangulation technique [23] is commonly used for this purpose but tends to be computationally expensive. Nevertheless, the Kinect sensors provide depth readings in a structured way (rectangular lattice). Advantage is taken of this structured information to perform the triangulation process efficiently using the following technique.

Let  $TR_m$  be the triangle mesh to be calculated for a given point cloud  $P_L$ . The latter being a rectangular lattice of 3D points under the constraint that Kinect sensors are used. Let  $h$  be a variable threshold that defines the surface continuity and

let  $pe$ ,  $ps$ ,  $pw$ ,  $pn$  be the four neighbors of a point  $pi$  defined as shown in Figure 17. The parameter  $h$  increases in accordance with the distance between the sensor and the surface due to the fact that the error of the sensor increases along the  $z$ -axis.

Now, let us assume that  $pi$ ,  $pe$ ,  $ps$ ,  $pw$ , and  $pn$  are indices of the point cloud vector and that a 3D point  $v_i$  can be accessed in this way:  $v_i \leftarrow P_L(\text{index})$ . If there is a hole in  $P_L$ , the  $z$  coordinates of the missing points are set to  $\infty$ . Given this structure, the triangle mesh generation is accelerated as follows.

For each point  $i$  in  $P_L$ , two candidate triangles are evaluated individually according to the continuity criteria of their vertices. The criterion for the inclusion or rejection of a triangle consists in comparing the threshold  $h$ , mentioned previously, to the difference of the  $z$ -coordinates between pairs of points. If the check is passed successfully, the three-sided polygon is added to the mesh. The triangles are defined as follows:  $T_{m_1}[j] \leftarrow \text{triangle}(v_1, v_2, v_3)$  and  $T_{m_2}[j] \leftarrow \text{triangle}(v_1, v_4, v_5)$ , where  $v_1 = P_L(i)$ ,  $v_2 = P_L(e)$ ,  $v_3 = P_L(s)$ ,  $v_4 = P_L(w)$ , and  $v_5 = P_L(n)$ . By using this technique, the triangulation process is performed rapidly. The downside of the method is the fact that it tends to leave small holes if there are missing points in the point cloud.

## 7. Experimental Evaluation

This section presents a series of experiments aimed to evaluate the proposed sensing framework. The first experiment assesses the tolerance of the sensors when they capture irregular shapes of car body parts over simulated uneven surfaces such as that in semicontrolled environments. The framework provides point clouds of the different sections of the vehicle; the information is aligned and it is used to create a triangle mesh of the surface. Thus, the next experiment evaluates the performance of the proposed mesh generation algorithm. This section closes with a qualitative evaluation of full 3D reconstructions of lateral panels of real vehicles and large objects.

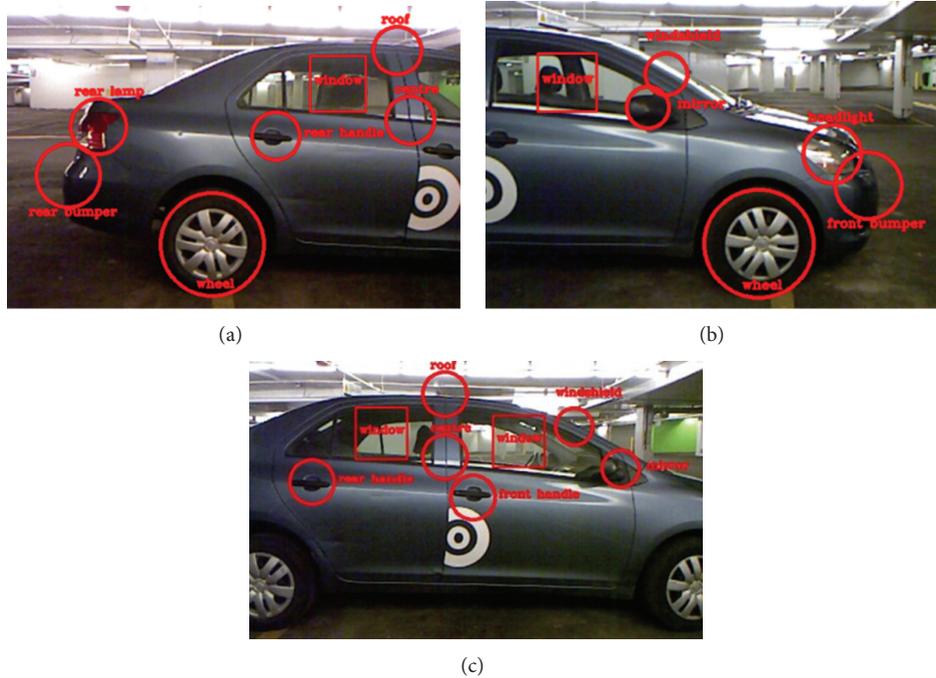


FIGURE 16: Automatic detection of fourteen parts of interest over three different RGB images which correspond to different sections of a vehicle: (a) view from sensor  $K_0$ , (b) sensor  $K_2$ , and (c) sensor  $K_1$ .

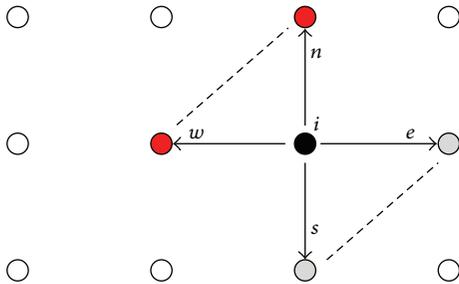


FIGURE 17: 2D representation of a 3D point cloud; the  $z$ -coordinate of each point was removed. The solid dot is a vertex  $i$  shared by the triangles  $(i, e, s)$  and  $(i, w, n)$ .

In order to evaluate the capability of the system to reconstruct irregular surfaces, a model of a car door was used, as shown in Figure 6(a) and was imaged with a single Kinect sensor. The model was designed for simulating the irregular curves of an actual vehicle surface as well as folds and pleats, as shown in Figure 18(a). A set of salient points over the surface of the door was selected to evaluate the accuracy of the method. Using these points, a silhouette of the panel is built. Figure 18(a) shows the salient points in both the door model and its reconstruction. Five points of the outline represent tight curves over the surface. Additionally, three small bumps were added over the smooth part of the surface to be used as reference points. Figure 18(b) presents the results of constructing the silhouettes for the door model using different point clouds. The data were collected with the sensor tilted at angles of 5, 10, and 15 degrees with respect to the horizontal level. The sensor was positioned at 30 cm above the bottom

of the door and 1.5 m from the closest point of the door. The uppermost points of the outlines were selected as a common point for the alignment of the silhouettes.

In Figure 18(b), the black line represents the actual shape of the model. Color lines represent results from the experiments under three different inclinations of the sensor with respect to the object. The error along the  $y$  and  $z$  coordinates, for most of the points, remains in the interval  $(0, \pm 1.8)$  cm with respect to the actual door model which proves that the scanning system achieves a good degree of tolerance to irregular leveling and alignment of the sensors. The image shown in Figure 19 is the 3D reconstruction and 3D mesh of the mock-up door. For visualization purposes in this paper, the Quadric Clustering decimation algorithm [24] was applied over the triangle mesh to reduce the number of triangles.

The next experiment was conducted to measure the performance of the proposed algorithm for triangle mesh generation. The results are compared, in Table 3, with those obtained by using the Delaunay algorithm provided by the Visualization Toolkit library (VTK) [25]. Sensors  $K_0$ ,  $K_1$ ,  $K_2$  were used to create three individual triangle meshes for their corresponding bodywork region. The results shown in Table 3 were obtained after running the algorithms 10 times over each point cloud. The difference in the results can be explained by the fact that the Delaunay algorithm is designed to work with nonstructured 3D points while the proposed technique from Section 6 takes advantage of the rectangular lattice of 3D points produced by the Kinect sensors.

Lastly, experiments were conducted using the proposed scanning system and real vehicles. In this test, a full reconstruction of the lateral panels of different vehicles was achieved to evaluate the capability of Kinect sensors to

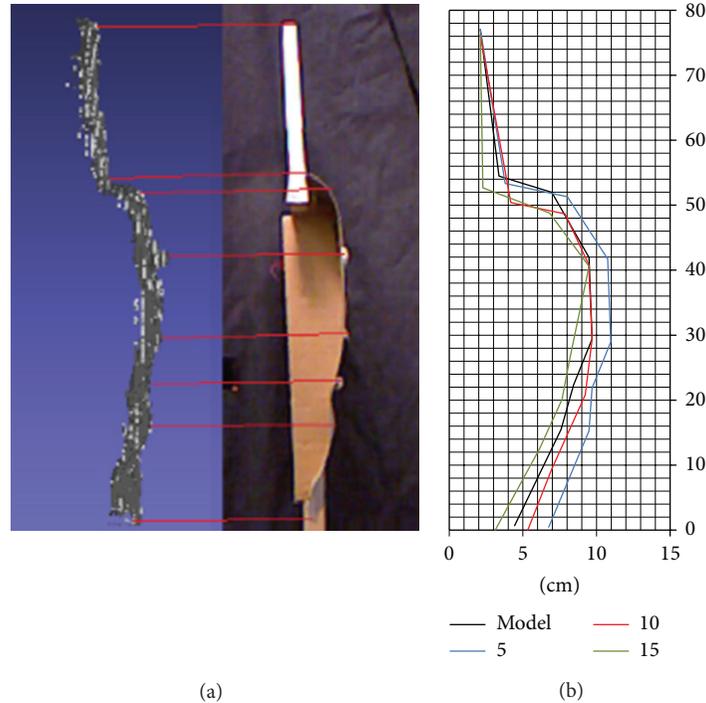


FIGURE 18: (a) Profile of a car door model over which a set of salient points were selected to construct the outline of the surface for evaluation purposes, and (b) silhouette reconstruction with the sensor tilted at different angles.

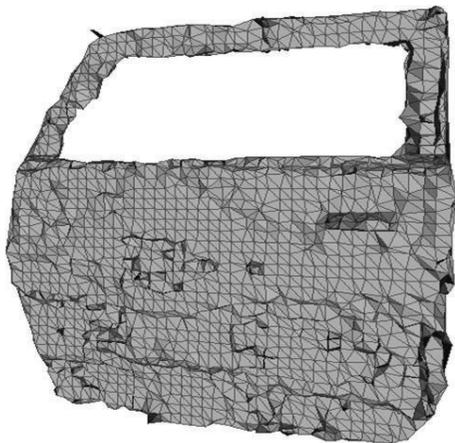


FIGURE 19: Triangle mesh of a mockup door model.

TABLE 3: Comparison of algorithms performance for triangle mesh generation.

Vehicle panel	Average number of vertices	Delaunay algorithm, average time	Proposed triangle mesh generation algorithm, average time
Front	174687	3.09 sec	0.086 sec
Central	219286	3.65 sec	0.171 sec
Back	217308	3.63 sec	0.138 sec

perform in semiconstrained environments and over large objects exhibiting complex surface reflectance characteristics.

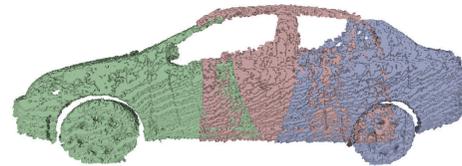


FIGURE 20: Point cloud registration of three different lateral sections of a 4-door sedan vehicle.

The first acquisition was performed in a semioutdoor parking garage over day time. Natural light was present in the scene via peripheral openings in the garage, while the sensors were protected from direct sunlight by means of the concrete ceiling. Figure 20 shows a sample of 3D reconstruction results obtained with the proposed methodology and a network of three Kinect sensors for the vehicle depicted in Figure 1(b). This acquisition was performed over winter season in Canada resulting in the vehicle’s side panels being covered with dirt and salt deposits from the road conditions, which created various shades of green paint, gray dirty areas, and specular reflection spots from the overhead lighting present in the installation. The raw information collected by each Kinect can be distinguished by its color for the front, central, and back panels. Figure 21 shows the complete 3D reconstruction results after registration and fusion based on the calibration phase performed with the proposed methodology. For visualization purposes a decimation algorithm was applied over the mesh such that triangles are large enough to be visible. This also impacts on the apparent smoothness of the displayed model. The 3D reconstruction is provided in the Stanford

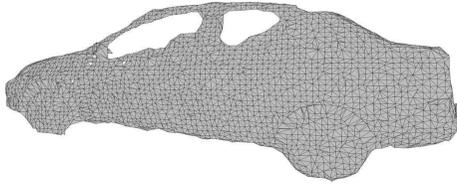


FIGURE 21: Resulting mesh for a Toyota Echo vehicle.



FIGURE 22: Capturing 3D data over a vehicle with the network of Kinect sensors.

Triangle Format [26], which is a convenient and standard way to represent the information of 3D meshes, and it can be used easily by a robot to plan its trajectory to approach the surface.

For the next experiment the acquisition was performed in an indoor parking garage. Figure 22 shows a vehicle standing in front of the experimental setup with 5 Kinect sensors for the rapid 3D modeling stage that will drive the robotic inspection. The scene was illuminated by incandescent lamps and a halogen lamp.

The reconstruction for two different types of vehicles is shown in Figures 23 and 24. In this case the texture captured by the Kinects' RGB cameras is added to the reconstructed scene, which provides a better appreciation of the reconstruction.

The windshield, lateral windows, and part of headlamp and rear lamp are missing in the depth maps because the IR energy generated by the Kinect devices passes through the transparent surfaces or is deflected in other directions. However, the rear window of the minivan, which is made of tinted glass, is partially captured. All of the main areas of the vehicles body and wheels, including dark rubber tires, are accurately reconstructed, and sections of the model acquired from the five viewpoints are correctly aligned, even over narrow roof supporting beams and highly curved bumper areas.

Table 4 presents a comparison between the characteristics of the reconstructed vehicle and their actual dimensions. The Kinect depth quantization introduces scaling errors of about 1 cm in height and width and a depth error of about 2.5 cm at 3 m distance. Each sensor covers the full height of the vehicle, and the average error on height is under 1%. The estimation of the length of the vehicle and the wheel base (i.e., the distance between the centers of the front and back wheels) involves all the calibration parameters estimated for the network of Kinect sensors. The error on the length is under 2.5%, which is relatively minor given the medium quality of data provided by



FIGURE 23: Six different views of a minivan vehicle 3D reconstruction.



FIGURE 24: Six different views of a semicompact vehicle 3D reconstruction.

TABLE 4: Reconstruction compared with ground truth.

	Height	Length	Wheel base
Car			
Actual (mm)	1460	4300	2550
Model (mm)	1471	4391	2603
Error (%)	<b>0.75</b>	<b>2.11</b>	<b>2.07</b>
Van			
Actual (mm)	1748	5093	3030
Model (mm)	1764	5206	3101
Error (%)	<b>0.91</b>	<b>2.21</b>	<b>2.34</b>

Kinect at a depth of 3 m and in proportion to the large working volume. For further assessment of the algorithm, an ICP algorithm [27] was applied on the point clouds, but it did not significantly improve the registration over what was achieved with the estimated calibration parameters. This confirms the accuracy of the initial calibration described in Section 4.

Finally, Figure 25 shows the reconstruction of other models of vehicles along with that of some garbage bins, acquired with the exact same setup, to evaluate the generalization capabilities of the proposed calibrated RGB-D acquisition framework. A wide range of vehicles was covered during experiments, in terms of colors and size. The white color vehicle appears more integrally than the vehicles with dark gray color, where missing depth data are noticed over the front part on the right of the vehicles where the density of points in the acquisition varies to a greater extent given the significant



FIGURE 25: Reconstruction of various vehicles and garbage bins.

change of alignment between Kinects  $K_2$  and  $K_3$ . The dark green garbage bins are also correctly reconstructed with proper alignment between the piecewise RGB-D models.

The algorithms presented in this paper were developed in C++ and run on a computer with an Intel core i7 CPU and Windows 7. The average time that the proposed method takes to reconstruct the 3D surface shape captured from each viewpoint for a regular vehicle is 4.0 sec. With regards to the acquisition time, the network of sensors collects the information in two time slots to avoid interference; the initialization of each device takes between 1 and 2 seconds. As a result, the scanning and 3D textured modeling processes for the entire vehicle are completed within 30 seconds. It is worth to say that most of that time is consumed by the subsystem for the visual detection of parts of interest. The calibration process is performed off-line, and the triangulation algorithm is run as the sensors are collecting the depth information.

The automated selection of regions of interest detailed in Section 5 allows for rapid extraction of subsets of the generated 3D model over which further processing can be performed, including higher resolution scanning, if required, over only limited but strategically selected surfaces in order to drive a robotic operation with higher precision over those regions. This proves an efficient strategy given that very high resolution acquisition and 3D modeling over an entire object of the size of a vehicle would be prohibitive in time and resources.

## 8. Conclusions

In this work, a rapid acquisition and reconstruction methodology for automated 3D modeling of large objects such as automotive vehicles is presented. The approach builds on a network of fast Kinect sensors distributed around the object to collect color and depth information over lateral panels of vehicles. The 3D modeling results are meant to provide a robotic arm with sufficiently accurate spatial information about the bodywork of a vehicle and the 3D location of up to fourteen features of interest over the surface such that it can interact with the automobile panels for various inspection or maintenance tasks.

This technology opens the door to a great number of real-time 3D reconstruction applications using low-cost RGB-D sensors. The main contributions of this work provide a reliable methodology to integrate multiple color and depth

streams of data in the task of 3D reconstruction of large objects. For that purpose, an efficient method for complete and accurate calibration of all intrinsic and extrinsic parameters of RGB-D sensor units specifically tailored to the Kinect sensors technology is presented. Furthermore, an approach for automated detection and recognition of critical regions of interest over vehicles from a mosaic of Kinect's color images is detailed. Finally, an accelerated triangular mesh generation algorithm is designed that takes advantage of the intrinsic structure of range data provided by Kinect sensors to further speed up the 3D model generation. The entire framework is experimentally validated under several operating conditions, including a laboratory environment, and semicontrolled parking garages where acquisition and 3D reconstruction are performed over objects of various sizes, including large vehicles. The results demonstrate the validity, accuracy, and rapidity of the use of Kinect's RGB-D sensors, in the context of robotic guidance. The addition of extra sensors to achieve full 360 degree coverage of a vehicle represents the next step of this investigation which will further extend the current capabilities.

## Conflict of Interests

The authors declare that there is no conflict of interests with any company or organization regarding the material discussed in this paper.

## Acknowledgments

The authors acknowledge the support from the Natural Sciences and Engineering Research Council of Canada to this research via the Strategic Project Grant no. STPGP 381229-09, as well as the collaboration of Scintrex Trace Corp.

## References

- [1] J. Zhou, J. Tong, L. Liu, Z. Pan, and H. Yan, "Scanning 3D full human bodies using kinects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 4, pp. 643–650, 2012.
- [2] A. Maimone and H. Fuchs, "Encumbrance-free telepresence system with real-time 3D capture and display using commodity depth cameras," in *Proceedings of the 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR '11)*, pp. 137–146, October 2011.
- [3] P. J. Noonan, T. F. Cootes, W. A. Hallett, and R. Hinz, "The design and initial calibration of an optical tracking system using the microsoft kinect," in *Proceedings of the IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC '11)*, pp. 3614–3617, October 2011.
- [4] P. Rakprayoon, M. Ruchanurucks, and A. Coundoul, "Kinect-based obstacle detection for manipulator," in *Proceedings of the IEEE/SICE International Symposium on System Integration (SII '11)*, pp. 68–73, 2011.
- [5] K. Berger, K. Ruhl, M. Albers et al., "The capturing of turbulent gas flows using multiple kinects," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV '11)*, pp. 1108–1113, Barcelona, Spain, November 2011.
- [6] J. Smisek, M. Jancosek, and T. Pajdla, "3D with kinect," in *Proceedings of the IEEE International Conference on Computer*

- Vision Workshops (ICCV '11)*, pp. 1154–1160, Barcelona, Spain, November 2011.
- [7] C.-S. Park, S.-W. Kim, D. Kim, and S.-R. Oh, “Comparison of plane extraction performance using laser scanner and kinect,” in *Proceedings of the 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI '11)*, pp. 153–155, Seoul, Korea, November 2011.
- [8] N. Burrus, “Demo software to visualize, calibrate and process kinect cameras output,” 2012, <http://nicolas.burrus.name/index.php/Research/KinectRgbDemoV6>.
- [9] M. Gaffney, “Kinect/3D scanner calibration pattern,” 2011, <http://www.thingiverse.com/thing:7793>.
- [10] K. Berger, K. Ruhl, Y. Schroeder, C. Brummer, A. Scholz, and M. Magnor, “Markerless motion capture using multiple color-depth sensors,” in *Proceedings of the Vision, Modeling and Visualization*, pp. 317–324, 2011.
- [11] K. Khoshelham, “Accuracy analysis of kinect depth data,” in *Proceedings of the ISPRS Workshop on Laser Scanning*, pp. 1437–1454, 2011.
- [12] S. Matyunin, D. Vatolin, Y. Berdnikov, and M. Smirnov, “Temporal filtering for depth maps generated by kinect depth camera,” in *Proceedings of the 5th 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON '11)*, pp. 1–4, May 2011.
- [13] S.-M. Kim, Y.-C. Lee, and S.-C. Lee, “Vision based automatic inspection system for nuts welded on the support hinge,” in *Proceedings of the SICE-ICASE International Joint Conference*, pp. 1508–1512, October 2006.
- [14] S. Agarwal, A. Awan, and D. Roth, “Learning to detect objects in images via a sparse, part-based representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1475–1490, 2004.
- [15] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff, “Semantic annotation, indexing, and retrieval,” *Journal of Web Semantics*, vol. 2, no. 1, pp. 49–79, 2004.
- [16] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, pp. 1511–1518, Kauai, Hawaii, USA, December 2001.
- [17] Y.-F. Fung, H. Lee, and M. F. Ercan, “Image processing application in toll collection,” *IAENG International Journal of Computer Science*, vol. 32, pp. 473–478, 2006.
- [18] M. M. Trivedi, T. Gandhi, and J. McCall, “Looking-in and looking-out of a vehicle: computer-vision-based enhanced vehicle safety,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 108–120, 2007.
- [19] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [20] Open Source Computer Vision Library, <http://opencv.willowgarage.com/wiki/>.
- [21] Real-time Computer Graphics and Physics, Mathematics, Geometry, Numerical Analysis, and Image Analysis, “Geometrictools,” <http://www.geometrictools.com/LibMathematics/Approximation/Approximation.html>.
- [22] A. Chávez-Aragón, R. Laganière, and P. Payeur, “Vision-based detection and labelling of multiple vehicle parts,” in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pp. 1273–1278, Washington, DC, USA, 2011.
- [23] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars, “Delaunay triangulations: height interpolation,” in *Computational Geometry: Algorithms and Applications*, pp. 191–218, Springer, 3rd edition, 2008.
- [24] P. Lindstrom, “Out-of-core simplification of large polygonal models,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*, pp. 259–262, July 2000.
- [25] The Visualization Toolkit, VTK, <http://www.vtk.org/>.
- [26] Stanford Triangle Forma, PLY, <http://www.cs.virginia.edu/~gfx/Courses/2001/Advanced.spring.01/plylib/Ply.txt>.
- [27] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, February 1992.

## Research Article

# Dynamic Compensation for Two-Axis Robot Wrist Force Sensors

**Junqing Ma, Aiguo Song, and Dongcheng Pan**

*Jiangsu Key Lab of Remote Measurement and Control, School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China*

Correspondence should be addressed to Aiguo Song; [a.g.song@seu.edu.cn](mailto:a.g.song@seu.edu.cn)

Received 10 June 2013; Accepted 26 June 2013

Academic Editor: Guangming Song

Copyright © 2013 Junqing Ma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To improve the dynamic characteristic of two-axis force sensors, a dynamic compensation method is proposed. The two-axis force sensor system is assumed to be a first-order system. The operation frequency of the system is expanded by a digital filter with backward difference network. To filter high-frequency noises, a low-pass filter is added after the dynamic compensation network. To avoid overcompensation, parameters of the proposed dynamic compensation method are defined by trial and error. Step response methods are utilized in dynamic calibration experiments. Compared to experiment data without compensation, the response time of the dynamic compensated data is reduced by 30%~40%. Experiments results demonstrate the effectiveness of our method.

## 1. Introduction

Multi-axis robot wrist force sensors are necessary for robotic systems in which contact force information between robots and environments needs to be obtained. There are various kinds of multi-axis force sensors available in commercial and research area, for example, cross-beam type multi-axis force sensors [1, 2], piezoelectric multi-axis force sensors [3], fiber multi-axis force sensors [4], and so on [5]. Multi-axis robot wrist force sensors are always mounted on the wrists of robots to convert multidimensional contact force signals into multi-channel voltage signals. Such kinds of applications can be frequently found in assemble robots, teleoperation robotic systems, rehabilitation robots, and so forth [6–9].

During a robot task, the effectiveness of on-line force perception and feedback highly relies on the performances of the multi-axis robot wrist force sensor. The strong real time and rapidity in robot tasks require multi-axis force sensors to perform high dynamic characteristic. However, multi-axis force sensors (hereafter referred to as “force sensors”) always have low natural frequency and small damping ratio owing to the low stiffness of elastic body and using of strain gauges. As a result, the dynamic response of the force sensors is more than 0.2 ms, and the adjusting time is relatively long [10]. The A/D converters for force sensors will prolong the response time as well. The disparity of dynamic requirements from robotic

tasks and the current performances of force sensors motivate the need to improve dynamic characteristics of force sensors. Improving dynamic characteristic of force sensors by hardware is limited and costly. In the field of measurement, dynamic performances of sensors are often improved by algorithms. Hence, dynamic compensation algorithms need to be designed to improve dynamic behavior of force sensors. Altintas and Park in [11] designed dynamic compensation algorithms by a Kalman filter for a spindle-integrated force sensor. Xu and Li in [10, 12] designed dynamic compensation algorithms by functional link artificial neural network (FLANN) for six-axis wrist force/torque sensor. Yu et al. in [13] designed dynamic compensation algorithms by genetic neural network for robot wrist force sensor, but the artificial neural network is slow to converge and may be subjected to local minimum [14].

In this paper, we proposed a dynamic compensation method based on a digital filter with backward difference design. Because there are high-frequency noises in the output voltages of force sensors, a low-pass filter is added in the dynamic compensation system. Dynamic calibration experiments are conducted, in which step-response method is utilized. During the dynamic calibration experiments, a two-axis force sensor which is designed and fabricated in our lab is used. Step signals of force are generated, while corresponding step responses of the two-axis force sensor are recorded by

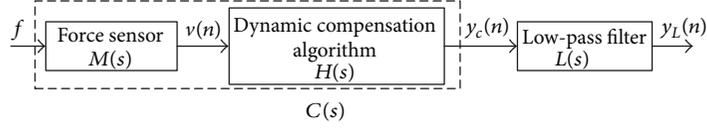


FIGURE 1: Flowchart of dynamic compensation of two-axis force sensors.

high-frequency data acquisition card. The proposed dynamic calibration method is applied to the dynamic calibration experiment data. The response time is greatly reduced with our method. The experiment results demonstrated the correctness and effectiveness of our method.

## 2. Dynamic Compensation System

**2.1. Principle of Dynamic Compensation.** The original operation frequency band of force sensors is too narrow to cover all the frequency components of input force signals. This causes the decays of high frequency components. Furthermore, dynamic behavior of force sensors will be decreased. In order to broaden the operation frequency, in this paper, a dynamic compensation part is added in force-sensor systems. With this method, dynamic performance of force sensors can be improved. Dynamic compensation of input forces in each dimension is done separately. Figure 1 shows the flowchart of the dynamic compensation of force sensors.

In Figure 1,  $f$  is the input force signal, and  $v(n)$  is the output voltage of the two-axis force sensors.  $y_c(n)$  is the results of  $v(n)$  passing dynamic compensation network, and  $y_L(n)$  is the dynamic compensated data after low-pass filter.  $M(s)$  represents the transfer function of the force sensor,  $H(s)$  represents the transfer function of the dynamic compensation,  $L(s)$  represents the transfer function of the low-pass filter, and  $C(s)$  represents the transfer function of the whole force sensor system.

We have

$$C(s) = M(s)H(s). \quad (1)$$

**2.2. Dynamic Compensation Algorithm.** The force sensor system, as shown in Figure 1, is assumed as a first-order system. The transfer function of the force sensor,  $M(s)$ , can be expressed as

$$M(s) = \frac{1}{1 + \tau s} e^{-\lambda s}, \quad (2)$$

where  $\tau$  is the time constant of the first-order system and  $\lambda$  is the lag time of the force sensor.

The cut-off angular frequency of the force sensor  $\omega$  can be calculated as

$$\omega = \frac{1}{\tau}. \quad (3)$$

If the frequency band of the force sensor is broadened to  $k$  times, the cut-off angular frequency becomes  $\omega_c = k\omega$ . Then,  $C(s)$  can be expressed as

$$C(s) = \frac{1}{1 + (\tau/k)s} e^{-\lambda s}. \quad (4)$$

Combinations of (1), (2), and (4) lead to

$$H(s) = \frac{C(s)}{M(s)} = \frac{1 + \tau s}{1 + (\tau/k)s}. \quad (5)$$

After A/D converter, the output voltages signals of force sensors are digital signals. Equivalent digital filter in  $z$  domain  $H(z)$  can be obtained from (5) and backward difference method as shown in (6):

$$s = \frac{1}{\tau_{\text{sam}}} (1 - z^{-1}), \quad (6)$$

where  $\tau_{\text{sam}}$  is the sampling interval.

Combination of (5) and (6) leads to

$$H(z) = k \frac{1 + c\tau_{\text{sam}} - z^{-1}}{1 + b\tau_{\text{sam}} - z^{-1}} = \frac{Y_c(z)}{X(z)}, \quad (7)$$

where  $c = 1/\tau$ ,  $b = k/\tau = kc$ .

The difference equation of (7) can be calculated as

$$y_c(n) = \frac{1}{1 + b\tau_{\text{sam}}} [k(1 + c\tau_{\text{sam}})x(n) - kx(n-1) + y_c(n-1)]. \quad (8)$$

Equation (8) is the dynamic compensation algorithm of force sensors. The proposed dynamic algorithm is simple and fast. It can be easily realized in software.

**2.3. Digital Low-Pass Filter.** As mentioned in the former section, the operation frequency of the force sensors is broadened in the proposed dynamic compensation algorithm. However, the extension of operation frequency will intensify high-frequency noises in the output voltage signals in each dimension. As a result, a low-pass filter is added after the dynamic compensation part, as shown in Figure 1, to filter out high frequency noise signals.

The common low-pass filter, called ‘‘moving average filter,’’ is utilized. Moving average filter is able to reduce random noises while retaining a sharp step response. As a result, moving average filter is the premier filter for time domain encoded signals with random noises. The moving average filter operates by averaging a number of points from input signals to produce every point in output signals [15].

In the dynamic compensation process, a moving average signal average  $m$  numbers of points in  $y_c(n)$ , as shown in (9):

$$y_L(n) = \frac{1}{m} \sum_{i=n-m+1}^n y_c(n+i), \quad n = 0, 1, 2, \dots \quad (9)$$



FIGURE 2: Prototype of a two-axis force sensor.

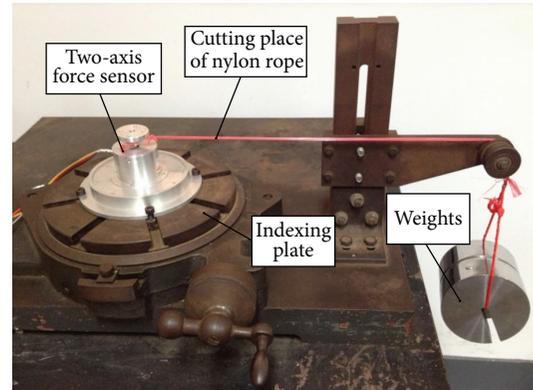


FIGURE 3: Dynamic compensation experiments setup.

### 3. Dynamic Compensation Results

**3.1. Dynamic Calibration Experiments.** A prototype two-axis force sensor which is designed and fabricated in our lab is used in the dynamic calibration experiments, as shown in Figure 2.

The key component of the force sensor is the cross-beam elastic body. When external forces are applied, the cross-beam elastic body will be deformed. Eight strain gauges which are pasted on the cross-beam elastic body of the force sensor will detect the deformation and convert the variations of input forces into variations of resistances. Each four strain gauges are connected to establish a Wheatstone bridge circuit. Variations of resistances of strain gauges can be converted into variations of voltages. The voltages are amplified by amplifying circuit [16].

The two-axis force sensor is able to measure horizontal forces in both X direction and Y direction. Because of the symmetrical characteristic of the two-axis force sensor, we only do dynamic calibration experiment in X direction.

There are three kinds of common dynamic calibration methods for sensors, namely, the frequency method, the step-response method, and the impact response method [12]. As for force sensors, a sine wave input force signal is difficult to generate. An impact force can be generated by a hammer with a piezoelectric sensor [17, 18]. However, the knocking position and angle are hard to control. Hence, step-response method [19] is preferred in dynamic calibration experiments for force sensors.

Figure 3 shows the platform of the dynamic calibration experiment setup.

As shown in Figure 3, a two-axis force sensor is mounted on an indexing plate in the center of the calibration table. The indexing plate can be rotated to ensure the directions of the loading forces. External horizontal forces are generated by pulley, nylon ropes and weights.

The cutting place of the nylon rope will be cut by a pair of scissors during the calibration experiment. When the nylon rope which is near the force sensor is cut, weights will fall down to the ground. A negative step force is generated. The cutting process should be quick and decisive.

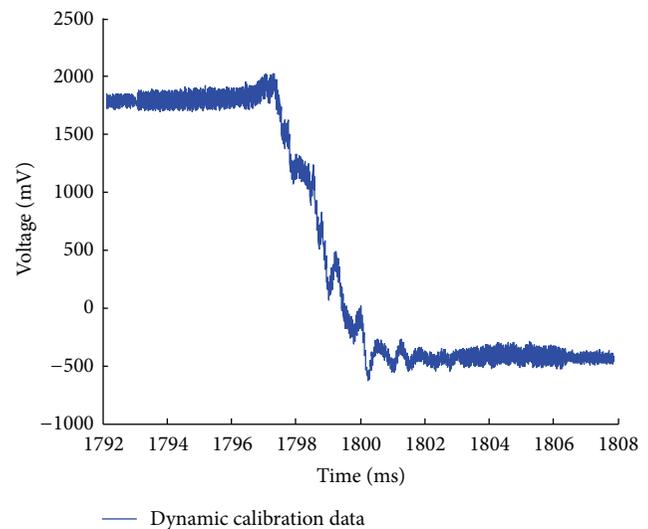


FIGURE 4: Original step response data from dynamic compensation experiments.

The response of the two-axis force sensor is recorded by a data acquisition card with a sampling frequency of 250 KHz.

**3.2. Results.** The whole dynamic calibration experiment is repeated three times. The best experiment data which show the fewest vibrations are chosen. Figure 4 shows the original negative step response data recorded by the data acquisition card.

The calibration experiment data support the assumption that the two-axis force sensor is a first-order system. The transfer function of the force sensor can be expressed in (2). The time constant  $\tau = 1$  ms, and the sampling interval  $\tau_{\text{sam}} = 0.004$  ms.

In the dynamic compensation network, parameters in the dynamic compensation algorithm and low-filter pass filter are defined by trial and error. Finally, the factor  $k$  is defined as 3; that is, the operation frequency is expanded three times.

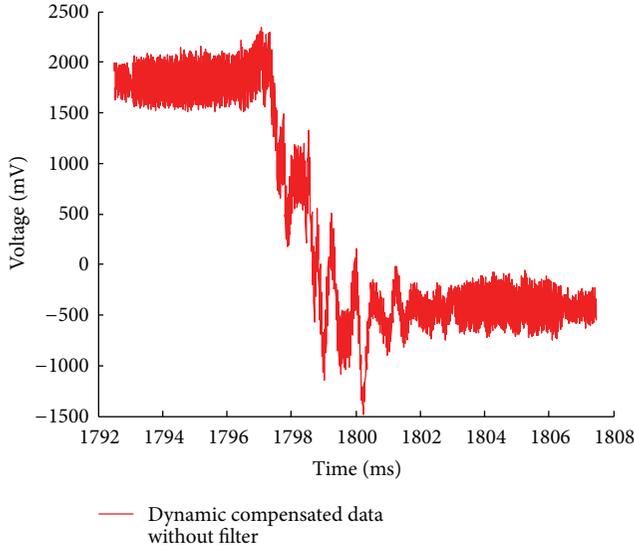


FIGURE 5: Dynamic compensation results of step response without low-pass filter.

The dynamic algorithm of the two-axis force sensor can be obtained as shown in (10):

$$y_c(n) = \frac{1}{1 + 0.012} \times [3 \times (1 + 0.004)v(n) - 3 \times v(n-1) + y_c(n-1)], \quad n = 1, 2, 3, \dots \quad (10)$$

Figure 5 shows dynamic compensated data without the low-pass filter. The red line indicates the step response data after dynamic compensation but without a low-pass filter.

Comparing Figure 5 with Figure 4, the red line indicates that much more high-frequency noises are induced from the dynamic compensation algorithm. As a result, a low-pass filter is necessary after the dynamic compensation network.

The moving average filter is utilized as the low-pass filter. The length of the moving average filter kernel is defined as 100, that is,  $m = 100$ . The low-pass filter can be expressed as

$$y_L(n) = \frac{1}{100} \sum_{i=n-50+1}^{n+50} y_c(n+i) \quad n = 0, 1, 2, \dots \quad (11)$$

In order to make a contrast, both the original data as shown in Figure 4 and the dynamic compensated data as shown in Figure 5 are passed through the moving average filter. Figure 6 shows comparisons between original data and dynamic compensated data after the low-pass filter.

In Figure 6, the black line represents original data filtered by the low-pass filter. The red line represents dynamic compensated data filtered by the same low-pass filter. Figure 6 shows that the response time of the two-axis force sensor is reduced from about 4 ms to about 2.5 ms. The time constant  $\tau$  is reduced from about 1 ms to about 0.6 ms.

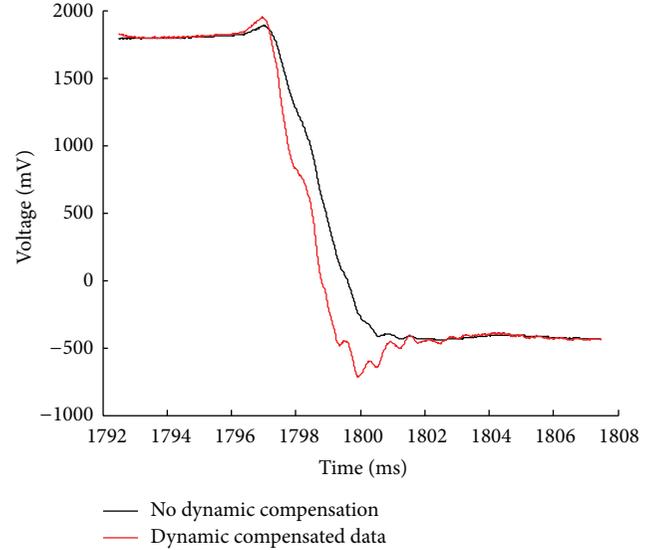


FIGURE 6: Comparisons between original data and dynamic compensated data after low-pass filter.

The unit step response function of the dynamic compensated two-axis force sensor can be expressed as

$$f_U(t) = \left(1 - e^{-(t-\lambda)/(6 \times 10^{-4})}\right) U(t - \lambda), \quad (12)$$

where  $U$  represents the unit step function.

As a result, the dynamic characteristic of the two-axis force sensor is improved a lot from the proposed dynamic compensation method. The dynamic compensation method is simple and not subjected to local minimum.

## 4. Conclusions

In this paper, a dynamic compensation method to efficiently improve dynamic characteristic of two-axis force sensors is presented. The dynamic compensation method contains a dynamic compensation network and a low-pass filter. The two-axis force sensor is assumed as a first-order system. The dynamic compensation network, which is based on backward difference method, broadens the operation frequency of the force sensors. High-frequency noises may be intensified from the compensation network. A moving average filter is added to filter the high-frequency noises.

Dynamic calibration experiments are implemented. A negative step force signal is generated by sudden release of applying loads. The step response obtained from the calibration experiment demonstrates the rationality of the assumption of force sensors to be a first-order system. The proposed dynamic compensation method is applied in the experiment data. Experiment results demonstrate the effectiveness of our method. The adjusting time in the step response is reduced from about 4 ms to about 2.5 ms. The time constant  $\tau$  is reduced from about 1 ms to about 0.6 ms.

Compared to the traditional dynamic compensation algorithm based on artificial neural network, our method is fast and not subjected to local minimum. The proposed dynamic

compensation method can also be implemented in other multi-axis force sensors.

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (no. 61272379).

## References

- [1] J. Ma and A. Song, "Fast estimation of strains for cross-beams six-axis force/torque sensors by mechanical modeling," *Sensors*, vol. 13, pp. 6669–6686, 2013.
- [2] A. Song, J. Wu, G. Qin, and W. Huang, "A novel self-decoupled four degree-of-freedom wrist force/torque sensor," *Measurement*, vol. 40, no. 9-10, pp. 883–891, 2007.
- [3] Y.-J. Li, B.-Y. Sun, J. Zhang, M. Qian, and Z.-Y. Jia, "A novel parallel piezoelectric six-axis heavy force/torque sensor," *Measurement*, vol. 42, no. 5, pp. 730–736, 2009.
- [4] M. S. Müller, L. Hoffmann, T. C. Buck, and A. W. Koch, "Fiber bragg grating-based force-torque sensor with six degrees of freedom," *International Journal of Optomechatronics*, vol. 3, no. 3, pp. 201–214, 2009.
- [5] R. J. Wood, K.-J. Cho, and K. Hoffman, "A novel multi-axis force sensor for microrobotics applications," *Smart Materials and Structures*, vol. 18, no. 12, Article ID 125002, 2009.
- [6] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, Springer, New York, NY, USA, 2008.
- [7] T. Lefebvre, J. Xiao, H. Bruyninckx, and G. De Gerssem, "Active compliant motion: a survey," *Advanced Robotics*, vol. 19, no. 5, pp. 479–499, 2005.
- [8] G. Xu, A. Song, and H. Li, "Adaptive impedance control for upper-limb rehabilitation robot using evolutionary dynamic recurrent fuzzy neural network," *Journal of Intelligent and Robotic Systems*, vol. 62, no. 3-4, pp. 501–525, 2011.
- [9] F. Nagata, Y. Kusumoto, K. Watanabe et al., "Polishing robot for PET bottle molds using a learning-based hybrid position/force controller," in *Proceedings of the 5th Asian Control Conference*, pp. 914–921, July 2004.
- [10] K.-J. Xu and C. Li, "Dynamic decoupling and compensating methods of multi-axis force sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 5, pp. 935–941, 2000.
- [11] Y. Altintas and S. S. Park, "Dynamic compensation of spindle-integrated force sensors," *CIRP Annals-Manufacturing Technology*, vol. 53, no. 1, pp. 305–308, 2004.
- [12] K.-J. Xu, C. Li, and Z.-N. Zhu, "Dynamic modeling and compensation of robot six-axis wrist force/torque sensor," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 5, pp. 2094–2100, 2007.
- [13] A. Yu, W. Huang, and G. Qin, "Dynamic modeling and compensation method based on genetic neural network for new type robot wrist force sensor," *Chinese Journal of Mechanical Engineering*, vol. 42, no. 12, pp. 239–244, 2006.
- [14] S. Chen and S. A. Billings, "Neural networks for nonlinear dynamic system modelling and identification," *International Journal of Control*, vol. 56, pp. 319–346, 1992.
- [15] P. J. Brockwell and R. A. Davis, *Time Series: Theory and Methods*, Springer, New York, NY, USA, 2009.
- [16] J. Ma and A. Song, "Development of a novel two-axis force sensor for chinese massage robot," *Applied Mechanics and Materials*, vol. 103, pp. 299–304, 2012.
- [17] N. Hu, H. Fukunaga, S. Matsumoto, B. Yan, and X. H. Peng, "An efficient approach for identifying impact force using embedded piezoelectric sensors," *International Journal of Impact Engineering*, vol. 34, no. 7, pp. 1258–1271, 2007.
- [18] M. Tracy and F.-K. Chang, "Identifying impact load in composite plates based on distributed piezoelectric sensor measurements," in *Smart Structures and Materials: Smart Structures and Integrated Systems*, vol. 2717 of *Proceedings of SPIE*, pp. 231–236, February 1996.
- [19] M. P. Schoen, "Dynamic compensation of intelligent sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 5, pp. 1992–2001, 2007.

## Research Article

# Modeling a Sensor to Improve Its Efficacy

Nabin K. Malakar,<sup>1</sup> Daniil Gladkov,<sup>2</sup> and Kevin H. Knuth<sup>3</sup>

<sup>1</sup> W. B. Hanson Center for Space Sciences, University of Texas at Dallas, Richardson, TX 75080, USA

<sup>2</sup> Department of Physics, University at Albany (SUNY), Albany, NY 12222, USA

<sup>3</sup> Departments of Physics and Informatics, University at Albany (SUNY), Albany, NY 12222, USA

Correspondence should be addressed to Nabin K. Malakar; nabinkm@gmail.com

Received 16 March 2013; Accepted 20 May 2013

Academic Editor: Guangming Song

Copyright © 2013 Nabin K. Malakar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Robots rely on sensors to provide them with information about their surroundings. However, high-quality sensors can be extremely expensive and cost-prohibitive. Thus many robotic systems must make due with lower-quality sensors. Here we demonstrate via a case study how modeling a sensor can improve its efficacy when employed within a Bayesian inferential framework. As a test bed we employ a robotic arm that is designed to autonomously take its own measurements using an inexpensive LEGO light sensor to estimate the position and radius of a white circle on a black field. The light sensor integrates the light arriving from a spatially distributed region within its field of view weighted by its spatial sensitivity function (SSF). We demonstrate that by incorporating an accurate model of the light sensor SSF into the likelihood function of a Bayesian inference engine, an autonomous system can make improved inferences about its surroundings. The method presented here is data based, fairly general, and made with plug-and-play in mind so that it could be implemented in similar problems.

## 1. Introduction

Robots rely on sensors to provide them with information about their surroundings. However, high-quality sensors can be cost-prohibitive and often one must make due with lower quality sensors. In this paper we present a case study which demonstrates how employing an accurate model of a sensor within a Bayesian inferential framework can improve the quality of inferences made from the data produced by that sensor. In fact, the quality of the sensor can be quite poor, but if it is known precisely how it is poor, this information can be used to improve the results of inferences made from the sensor data.

To accomplish this we rely on a Bayesian inferential framework where a machine learning system considers a set of hypotheses about its surroundings and identifies more probable hypotheses given incoming sensor data. Such inferences rely on a likelihood function, which quantifies the probability that a hypothesized situation could have given rise to the data. The likelihood is often considered to represent the noise model, and this inherently includes a model of how the sensor is expected to behave when presented with a given

stimulus. By incorporating an accurate model of the sensor, the inferences made by the system are improved.

As a test bed we employ an autonomous robotic arm developed in the Knuth Cyberphysics Laboratory at the University at Albany (SUNY). The robot is designed to perform studies in autonomous experimental design [1, 2]. In particular it performs autonomous experiments where it uses an inexpensive LEGO light sensor to estimate the position and radius of a white circle on a black field. The light sensor integrates the light arriving from a spatially distributed region within its field of view weighted by its spatial sensitivity function (SSF). We consider two models of the light sensor. The naïve model predicts that the light sensor will return one value on average if it is centered over a black region and another higher value on average if it is centered on a white region. The more accurate model incorporates information about the SSF of the light sensor to predict what values the sensor would return given a hypothesized surface albedo field. We demonstrate that by incorporating a more accurate model of the light sensor into the likelihood function of a Bayesian inference engine, a robot can make improved inferences about its surroundings. The efficacy of the sensor model

is quantified by the average number of measurements the robot needs to make to estimate the circle parameters within a given precision.

There are two aspects to this work. First is the characterization of the light sensor and second is the incorporation of the light sensor model into the likelihood function of the robot's machine learning system in a demonstration of improved efficacy. In Section 2 we describe the robot, its light sensor, the experiment that it is designed to perform, and the machine learning system employed. We then discuss the methods used to collect data from the light sensor, the models used to describe the light sensor, their incorporation into the machine learning system, and the methods used to estimate the model parameters and select the model order. Section 3 describes the resulting SSF model and the results of the experiments comparing the naïve light sensor model to the more accurate SSF model. In Section 4 we summarize our results which demonstrate how, by incorporating a more accurate model of sensor, one can improve its efficacy.

## 2. Materials and Methods

In this section we begin by discussing various aspects of the robotic test bed followed by a discussion of the techniques used to characterize the light sensor.

*2.1. Robotic Arm Test Bed.* The robotic arm is designed to perform studies in autonomous experimental design [1, 2]. The robot itself is constructed using the LEGO NXT Mindstorms system (Figure 1). The LEGO Mindstorms system was utilized in part to demonstrate that high-quality autonomous systems can be achieved when using lower quality equipment if the machine learning and data analysis algorithms are handled carefully. It employs one motor to allow it to rotate about a vertical axis indicated by the black line in the top center of the figure and two motors to extend and lower the arm about the joints located at the positions indicated by the short arrows. The motors are controlled directly by the LEGO brick, which is commanded via Bluetooth by a Dell Latitude laptop computer running the robot's machine learning system, which is programmed in MatLab (Mathworks, Inc.). The LEGO light sensor is attached to the end of the arm (indicated by the long arrow in Figure 1 and displayed in the inset at the upper right). Based on commands issued by the laptop, the robotic arm can deploy the light sensor to any position within its reach on the playing field. The light sensor is lowered to an average height of 14 mm above the surface before taking a measurement. The arm is designed using a trapezoidal construction that maintains the sensor's orientation to be aimed at nadir, always normal to the surface, despite the extension of the arm.

The LEGO light sensor (LEGO Part 9844) consists of a photodiode-LED pair. The white circle is the photo diode, and the red circle is the illuminating LED. Note that they are separated by a narrow plastic ridge, which prevents the LED from shining directly into the photo diode. This ridge, along with the plastic lenses and the presence of the illuminating LED, affects the spatial sensitivity of the sensor. When activated, the light sensor flashes for a brief instant

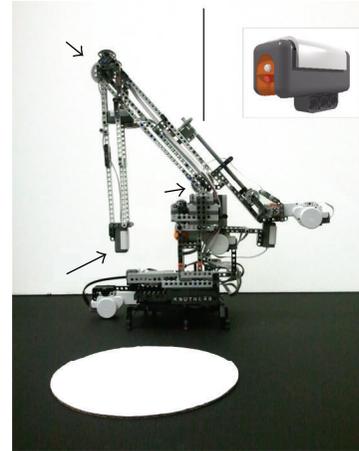


FIGURE 1: A photograph showing the robotic arm along with the circle it is programmed to characterize. The robotic arm is constructed using the LEGO NXT Mindstorms System. It employs one motor to allow it to rotate about a vertical axis indicated by the black line in the top center of the image and two motors to extend and lower the arm about the joints located at the positions indicated by the short arrows. The LEGO light sensor, also shown in the inset, is attached to the end of the arm as indicated by the long arrow.

and measures the intensity of the reflected light. The photodiode and its support circuitry are connected to the sensor port of the LEGO Brick (LEGO Part 9841), which runs on a 32 bit ARM7 ATMEEL microcontroller. The measured intensities are converted by internal software running on the ATMEEL microcontroller to a scale of 1 to 100, which we refer to as *LEGO units*. The light sensor integrates the light arriving from a spatially distributed region within its field of view. The spatial sensitivity of the sensor to light sources within its field of view is described by the spatial sensitivity function (SSF). This function is unknown, but if it were known, one could weight the surface albedo field with the SSF and integrate to obtain an estimate of the recorded intensity of the reflected light.

*2.2. The Circle Characterization Experiment.* The robotic arm is designed to deploy the light sensor to take measurements of the surface albedo at locations within a playing field of dimensions approximately  $131 \times 65$  LEGO distance units ( $1048 \text{ mm} \times 520 \text{ mm}$ ), within an automated experimental design paradigm [1–3]. The robot is programmed with a set of hypotheses of what shapes it could find placed on the playing field. Instead of being programmed with a specific set of strategies for characterizing the hypothesized shapes, the robot utilizes a generalized Bayesian Inference Engine coupled to an Inquiry Engine to make inferences about the hypotheses based on the recorded data and to use uncertainties in its inferences to drive further exploration by autonomously selecting a new measurement location that promises to provide the maximum amount of relevant information about the problem.

In this experiment, the robotic arm is instructed that there is a white circle of unknown radius arbitrarily placed on

the black field. Such an instruction is encoded by providing the robot with a model of the surface albedo consisting of three parameters: the center location  $(x_o, y_o)$  and radius  $r_o$ , written jointly as

$$\mathbf{C} = \{(x_o, y_o), r_o\}, \quad (1)$$

so that given a measurement location  $(x_i, y_i)$  the albedo  $S$  is expected to be

$$S(x_i, y_i; \mathbf{C}) = \begin{cases} 1 & \text{if } D((x_i, y_i), (x_o, y_o)) \leq r_o, \\ 0 & \text{if } D((x_i, y_i), (x_o, y_o)) > r_o, \end{cases} \quad (2)$$

where

$$D((x_i, y_i), (x_o, y_o)) = \sqrt{(x_i - x_o)^2 + (y_i - y_o)^2} \quad (3)$$

is the Euclidean distance between the measurement location  $(x_i, y_i)$  and the center of the circle  $(x_o, y_o)$ , and an albedo of 1 signifies that the surface is white and 0 signifies that the surface is black. Precisely how these expectations are used to make inferences from data is explained in the next section. Keep in mind that while the circle's precise radius and position are unknown, the robot has been provided with limited prior information about the allowable range of radii and positions.

Again, it is important to note that the robot does not scan the surface to solve the problem nor does it try to find three points along the edge of the circle. Instead, it employs a general system that works for any expected shape or set of shapes that autonomously and intelligently determines optimal measurement locations based both on what it knows and on what it does not know. The number of measurements needed to characterize all three circle parameters within the desired accuracy is a measure of the efficiency of the experimental procedure.

**2.3. The Machine Learning System.** The machine learning system employs a Bayesian Inference Engine to make inferences about the circle parameters given the recorded light intensities, as well as an Inquiry Engine designed to use the uncertainties in the circle parameter estimates to autonomously select measurement locations that promise to provide the maximum amount of relevant information about the problem.

The core of the Bayesian Inference Engine is centered around the computation of the posterior probability  $\Pr(\mathbf{C} | \mathbf{D}, I)$  of the albedo model parameters,  $\mathbf{C}$  in (1) given the light sensor recordings (data)  $d_i$  recorded at locations  $(x_i, y_i)$ , which we write compactly as

$$\mathbf{D} = \{(d_1, (x_1, y_1)), \dots, (d_N, (x_N, y_N))\} \quad (4)$$

and any additional prior information  $I$ . Bayes' Theorem allows one to write the posterior probability as a function of three related probabilities

$$\Pr(\mathbf{C} | \mathbf{D}, I) = \Pr(\mathbf{C} | I) \frac{\Pr(\mathbf{D} | \mathbf{C}, I)}{\Pr(\mathbf{D} | I)}, \quad (5)$$

where the right-hand side consists of the product of the prior probability of the circle parameters  $\Pr(\mathbf{C} | I)$ , which describes what is known about the circle before any sensor data are considered, with a ratio of probabilities that are sensor data dependent. It is in this sense that Bayes' Theorem represents a learning algorithm since what is known about the circle parameters before the data are considered (prior probability) is modified by the recorded sensor data resulting in a quantification of what is known about the circle parameters after the data are considered (posterior probability). The probability in the numerator on the right is the likelihood  $\Pr(\mathbf{D} | \mathbf{C}, I)$ , which quantifies the probability that the sensor data could have resulted from the hypothesized circle parameterized by  $\mathbf{C}$ . The probability in the denominator is the *marginal likelihood* or the *evidence*, which here acts as a normalization factor. Later when estimating the SSF of the light sensor (which is a different problem), the evidence, which can be written as the integral of the product of the prior and the likelihood over all possible model parameters, will play a critical role.

The likelihood term,  $\Pr(\mathbf{D} | \mathbf{C}, I)$ , plays a critical role in the inference problem, since it essentially compares predictions made using the hypothesized circle parameters to the observed data. A naïve light sensor model would predict that if the sensor was centered on a black region, it would return a small number, and if the sensor was centered on a white region, it would return a large number. A more accurate light sensor model would take into account the SSF of the light sensor and perform an SSF-weighted integral of the hypothesized albedo field and compare this to the recorded sensor data. These two models of the light sensor will be discussed in detail in the next section.

The robot not only makes inferences from data, but also designs its own experiments by autonomously deciding where to take subsequent measurements [1–6]. This can be viewed in terms of Bayesian experimental design [7–13] where the Shannon entropy [14] is employed as the utility function to decide where to take the next measurement. In short, the Bayesian Inference Engine samples 50 circles from the posterior probability using the nested sampling algorithm [15, 16]. Since these samples may consist of replications, they are further diversified by taking approximately 1000 Metropolis-Hastings steps [17, 18]. Figure 2 illustrates the robot's machine learning system's view of the playing field after several measurements have been recorded. The 50 circles sampled from the posterior probability (blue circles) reflect what the robot knows about the white circle (not shown) given the light sensor data that it has collected. The algorithm then considers a fine grid of potential measurement locations on the playing field. At each potential measurement location, the 50 sampled circles are queried using the likelihood function to produce 50 samples of what measurement could be expected at that location given each hypothesized circle. This results in a set of potential measurement values at each measurement location. The Shannon entropy of the set of potential measurements at each location is computed, which results in an entropy map, which is illustrated in Figure 2 as the copper-toned coloration upon which the blue circles lie. The set of measurement locations with the greatest entropy

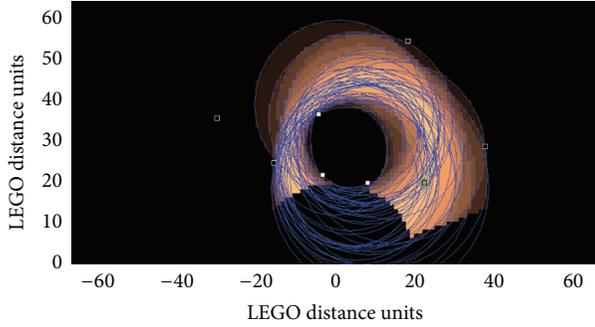


FIGURE 2: This figure illustrates the robot's machine learning system's view of the playing field using the naïve light sensor model. The axes label playing field coordinates in LEGO distance units. The previously obtained measurement locations used to obtain light sensor data are indicated by the black and white squares indicating the relative intensity with respect to the naïve light sensor model. The next selected measurement location is indicated by the green square. The blue circles represent the 50 hypothesized circles sampled from the posterior probability. The shaded background represents the entropy map, such that brighter areas indicate the measurement locations that promise to provide greater information about the circle to be characterized. Note that the low entropy area bounded by the white squares indicates that this region is probably inside the white circle and that measurements made here will not be as informative as measurements made elsewhere. The dark jagged edges at the bottom of the colored high entropy regions reflect the boundary between the playing field and the region that is outside of the robotic arm's reach.

are identified, and the next measurement location is randomly chosen from that set. Thus the likelihood function not only affects the inferences about the circles made by the machine learning algorithm, but also affects the entropy of the measurement locations, which guides further exploration. In the next section, we describe the two models of the light sensor and their corresponding likelihood functions.

The efficacy of the sensor model will be quantified by the average number of measurements the robot needs to make to estimate the circle parameters within a given precision.

**2.4. Models Describing a Light Sensor.** In this section we discuss two models of a light sensor and indicate precisely how they are integrated into the likelihood function used by both the Bayesian Inference and Inquiry Engines.

**2.4.1. The Naïve Likelihood.** A naïve light sensor model would predict that if the sensor was centered on a black region (surface albedo of zero), the sensor would return a small number on average, and if it were centered on a white region (surface albedo of unity), it would return a large number on average. Of course, there are expected to be noise variations from numerous sources, such as uneven lighting of the surface, minor variations in albedo, and noise in the sensor itself. So one might expect that there is some expected squared deviation from the two mean sensor values for the white and black cases. For this reason, we model the expected sensor response with a Gaussian distribution with mean  $\mu_B$  and standard deviation  $\sigma_B$  for the black surface and a Gaussian

distribution with mean  $\mu_W$  and standard deviation  $\sigma_W$  for the white surface. The likelihood of a measurement  $d_i$  at location  $(x_i, y_i)$  corresponding to the naïve light sensor model,  $\Pr_{\text{naive}}(\{(d_i, (x_i, y_i))\} | \mathbf{C}, I)$ , can be written compactly as

$$\Pr_{\text{naive}}(\{(d_i, (x_i, y_i))\} | \mathbf{C}, I) = \begin{cases} (2\pi\sigma_W)^{-1/2} \exp\left[-\frac{(\mu_W - d_i)^2}{2\sigma_W^2}\right] & \text{for } D((x_i, y_i), (x_o, y_o)) \leq r_o, \\ (2\pi\sigma_B)^{-1/2} \exp\left[-\frac{(\mu_B - d_i)^2}{2\sigma_B^2}\right] & \text{for } D((x_i, y_i), (x_o, y_o)) > r_o, \end{cases} \quad (6)$$

where  $\mathbf{C} = \{(x_o, y_o), r_o\}$  represents the parameters of the hypothesized circle and  $D((x_i, y_i), (x_o, y_o))$  is the Euclidean distance given in (3). The joint likelihood for  $N$ -independent measurements is found by taking the product of the  $N$  single-measurement likelihoods. In a practical experiment, the means  $\mu_B$  and  $\mu_W$  and standard deviations  $\sigma_B$  and  $\sigma_W$  can be easily estimated by sampling known black and white regions several times using the light sensor.

**2.5. The SSF Likelihood.** A more accurate likelihood can be developed by taking into account the fact that the photo diode performs a weighted integral of the light arriving from a spatially distributed region within its field of view, the weights being described by the spatial sensitivity function (SSF) of the sensor. Since the SSF of the light sensor could be arbitrarily complex with many local peaks, but is expected to decrease to zero far from the sensor, we characterize it using a mixture of Gaussians (MoG) model, which we describe in this section.

The sensor's response situated a fixed distance above the point  $(x_i, y_i)$  to a known black-and-white pattern which can be modeled in the lab frame by

$$M(x_i, y_i) = I_{\min} + (I_{\max} - I_{\min})R(x_i, y_i), \quad (7)$$

where  $I_{\min}$  and  $I_{\max}$  are observed intensities for a completely black surface (surface albedo of zero) and a completely white surface (surface albedo of unity), respectively, and  $R$  is a scalar response function, varying between zero and one that depends both on the SSF and the surface albedo [19]. The minimum intensity  $I_{\min}$  acts as an offset and  $(I_{\max} - I_{\min})$  serves to scale the response to LEGO units.

The response of the sensor is described by  $R(x_i, y_i)$ , which is a convolution of the sensor SSF and the surface albedo  $S(x, y)$  given by

$$R(x_i, y_i) = \int dx dy \text{SSF}(x - x_i, y - y_i) S(x, y), \quad (8)$$

where the SSF is defined so that the convolution with a completely white surface results in a response of unity. In practice, we approximate this integral as a sum over a pixelated grid with 1 mm square pixels

$$R(x_i, y_i) = \sum_{x,y} \text{SSF}(x - x_i, y - y_i) S(x, y). \quad (9)$$

We employ a mixture of Gaussians (MOG) as a parameterized model to describe the SSF in the sensor's frame coordinates  $(x', y') = (x - x_i, y - y_i)$

$$\text{SSF}(x', y') = \frac{1}{K} \sum_{n=1}^N a_n \times \exp \left[ - \left\{ A_n (x' - u'_n)^2 + B_n (y' - v'_n)^2 + 2C_n (x' - u'_n) (y' - v'_n) \right\} \right], \quad (10)$$

where  $(u'_n, v'_n)$  denotes the center of the  $n$ th two-dimensional Gaussian with amplitude  $a_n$  and covariance matrix elements given by  $A_n, B_n$  and  $C_n$ . The constant  $K$  denotes a normalization factor, which ensures that the SSF integrates to unity [19]. The model is sufficiently general so that one could vary the number of Gaussians to any number, although we found that in modeling this light sensor testing models with  $N$  varying from  $N = 1$  to  $N = 4$  was sufficient. The MoG model results in a set of six parameters to be estimated for each Gaussian

$$\theta_n = \{a_n, u'_n, v'_n, A_n, B_n, C_n\}, \quad (11)$$

where the subscript  $n$  is used to denote the  $n$ th Gaussian in the set. These must be estimated along with the two intensity parameters,  $I_{\min}$  and  $I_{\max}$  in (7) so that an MoG model with  $N$  Gaussians consists of  $6N + 2$  parameters to be estimated.

We assign a Student- $t$  distribution to the SSF model likelihood, which can be arrived at by assigning a Gaussian likelihood and integrating over the unknown standard deviation  $\sigma$  [16]. By defining  $\mathbf{D} = \{(d_1, (x_1, y_1)), (d_2, (x_2, y_2)), \dots, (d_N, (x_N, y_N))\}$  and writing the hypothesized circle parameters as  $\mathbf{C} = \{(x_o, y_o), r_o\}$ , we have

$$\text{Pr}_{\text{SSF}}(\mathbf{D} | \mathbf{C}, I) \propto \left[ \sum_{i=1}^N (M(x_i, y_i) - d_i)^2 \right]^{-(N-1)/2}, \quad (12)$$

where  $N$  is the number of measurements made by the robot, and the function  $M$ , defined in (7), relies on both (8) and (10). Note that in practice, the MoG SSF model described in (10) is used to generate a discrete SSF matrix which is used in the convolution (9) to compute the likelihood function via the sensor response model  $M$  in (7).

**2.6. Data Collection for SSF Estimation.** In this section we describe the collection of the light sensor readings in the laboratory that were used to estimate the SSF of the light sensor. The SSF is a function not only of the properties of the photodiode, but also of the illuminating LED and the height above the surface. For this reason, measurements were made at a height of 14 mm above a surface with a known albedo in a darkened room to avoid complications due to ambient light and to eliminate the possibility of shadows cast by the sensor or other equipment [19].

The surface, which we refer to as the black-and-white boundary pattern, consisted of two regions: a black region on the left and a white region on the right separated by a sharp

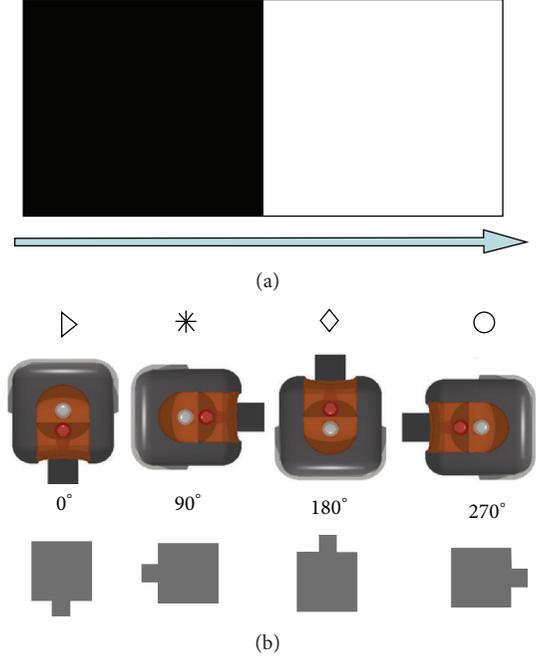


FIGURE 3: (a) This figure illustrates the laboratory surface, referred to as the black-and-white boundary pattern, with known albedo, which consisted of two regions: a black region on the left and a white region on the right separated by a sharp linear boundary. (b) This illustrates the four sensor orientations used to collect data for the estimation of the SSF along with the symbols used to indicate the measured values plotted in Figure 5. The top row views the sensors as if looking up from the table so that the photodiode placement in the sensor package can be visualized. Below these the gray sensor shapes illustrate how they are oriented looking down at both the sensors and the albedo surface. Measurements were taken as the sensor was incrementally moved, with one-millimeter steps, in a direction perpendicular to the boundary (as indicated by the arrow at the bottom of the figure) from a position of 5 cm to the left of the boundary (well within the completely black region) to a position 5 cm to the right of the boundary (well within the completely white region). This process was repeated four times with the sensor in each of the four orientations.

linear boundary as shown in Figure 3(a). Here the surface and sensor are illustrated as if viewing them by looking up at them from below the table surface, so that the placement of the photodiode in the sensor package can be visualized. The lab frame was defined to be at the center of the black-white boundary so that the surface albedo,  $S_{\text{BW}}(x, y)$ , is given by

$$S_{\text{BW}}(x, y) = \begin{cases} 1, & \text{for } x > 0, \\ 0, & \text{for } x \leq 0. \end{cases} \quad (13)$$

Measurements were taken as the sensor was incrementally moved, with one-millimeter steps, in a direction perpendicular to the boundary from a position of 5 cm to the left of the boundary (well within the completely black region) to a position 5 cm to the right of the boundary (well within the completely white region) resulting in 101 measurements. This process was repeated four times with the sensor in each of

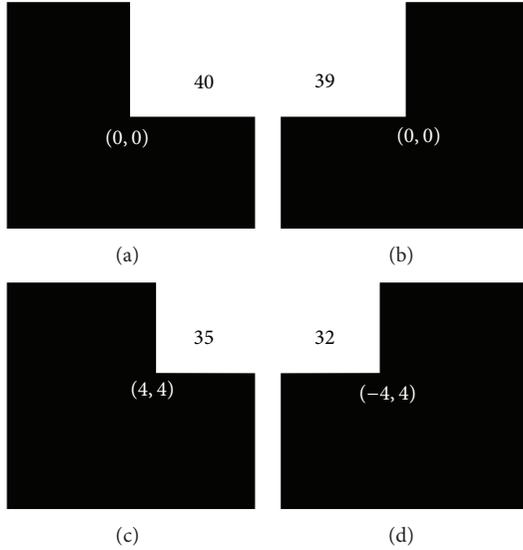


FIGURE 4: Four additional symmetry-breaking albedo patterns were employed. In all cases, the sensor was placed in the  $0^\circ$  orientation at the center of the pattern, indicated by  $(0,0)$ . In the two lower patterns, the center of the white square was shifted diagonally from the center by 4 mm as indicated by the coordinates (white text) of the corner. The recorded intensity levels are displayed in the white albedo area.

the four orientations (see Figure 3 for an explanation), giving a total of 404 measurements using this pattern.

The black-and-white boundary pattern does not provide sufficient information to uniquely infer the SSF, since the sensor may have a response that is symmetric about a line oriented at  $45^\circ$  with respect to the linear boundary. For this reason, we employed four additional albedo patterns consisting of black regions with one white quadrant as illustrated in Figure 4, which resulted in four more measurements. These have surface albedos defined by

$$\begin{aligned}
 S_a(x, y) &= \begin{cases} 1, & \text{for } x > 0 \text{ mm, } y > 0 \text{ mm,} \\ 0, & \text{otherwise,} \end{cases} \\
 S_b(x, y) &= \begin{cases} 1, & \text{for } x < 0 \text{ mm, } y > 0 \text{ mm,} \\ 0, & \text{otherwise,} \end{cases} \\
 S_c(x, y) &= \begin{cases} 1, & \text{for } x > 4 \text{ mm, } y > 4 \text{ mm,} \\ 0, & \text{otherwise,} \end{cases} \\
 S_d(x, y) &= \begin{cases} 1, & \text{for } x < -4 \text{ mm, } y > 4 \text{ mm,} \\ 0, & \text{otherwise,} \end{cases}
 \end{aligned} \tag{14}$$

where the subscripts relate each albedo function to the pattern illustrated in Figure 4.

**2.7. Estimating SSF MoG Model Parameters.** In this section we describe the application of Bayesian methods to estimate the SSF MoG model parameters. Keep in mind that in this paper we are considering two distinct inference problems: the robot's inferences about a circle and our inferences about

the light sensor SSF. Both of these problems rely on making predictions about measured intensities using a light sensor. For this reason many of these equations will not only look similar to what we have presented previously, but also depend on the same functions mapping modeled albedo fields to predicted sensor responses, which are collectively represented using the generic symbol  $\mathbf{D}$ . It may help to keep in mind that the robot is characterizing a circle quantified by model parameters represented jointly by the symbol  $\mathbf{C}$ , and we are estimating the SSF of a light sensor quantified by model parameters represented jointly by the symbol  $\theta$  below. With the exception of the evidence, represented by function  $Z$  below (which is not used by the robot in this experiment), all of the probability functions contain the model parameters in their list of arguments making it clear to which inference problem they refer.

The posterior probability for the SSF MoG model parameters, collectively referred to as  $\theta = \{\theta_1, \theta_2, \dots, \theta_N\}$ , for a model consisting of  $N$  Gaussians, is given by

$$\Pr(\theta \mid \mathbf{D}, I) = \frac{1}{Z} \Pr(\theta \mid I) \Pr(\mathbf{D} \mid \theta, I), \tag{15}$$

where here  $\mathbf{D}$  refers to the data collected for the SSF estimation experiment described in the previous section,  $I$  refers to our prior information about the SSF (which is that it may have several local peaks and falls off to zero far from the sensor), and  $Z$  refers to the evidence  $Z = \Pr(\mathbf{D} \mid I)$ , which can be found by

$$Z = \int d\theta \Pr(\theta \mid I) \Pr(\mathbf{D} \mid \theta, I). \tag{16}$$

In our earlier discussion where Bayes' theorem was used to make inferences about circles, the evidence played the role of a normalization factor. Here, since we can consider MoG models with different numbers of Gaussians and since we integrate over all of the possible values of the parameters, the evidence quantifies the degree to which the hypothesized model order  $N$  supports the data. That is, the optimal number of Gaussians to be used in the MoG model of the SSF can be found by computing the evidence.

All five sets of data  $\mathbf{D}$  described in the previous section were used to compute the posterior probability. These are each indexed by the subscript  $i$  where  $i = 1, 2, 3, 4$ , so that  $D_i$  refers to the data collected using each of the four orientations  $\phi_i = \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$  to scan the black-and-white boundary pattern resulting in  $N_i = 101$  measurements for  $i = 1, 2, 3, 4$ , and where the value  $i = 5$  refers to the  $N_i = 4$  measurements attained using the  $0^\circ$  orientation with the set of four additional patterns.

We assign uniform priors so that this is essentially a maximum likelihood calculation with the posterior being proportional to the likelihood. We assign a Student- $t$  distribution to the likelihood, which can be arrived at by assigning

a Gaussian likelihood and integrating over the unknown standard deviation  $\sigma$  [16]. This can be written as

$$\Pr(D_i | \theta, I) \propto \left[ \sum_{j=1}^{N_i} (M_{ij}(x_{ij}, y_{ij}) - D_i(x_{ij}, y_{ij}))^2 \right]^{-(N_i-1)/2}, \quad (17)$$

where  $i$  denotes each of the five sets of data and  $j$  denotes the  $j$ th measurement of that set, which was taken at position  $(x_{ij}, y_{ij})$ . The function  $M_{ij}(x, y)$  represents a predicted measurement value obtained from (7) using (9) with the albedo function  $S(x, y)$  defined using the albedo pattern  $S_{\text{BW}}(x, y)$  with orientation  $\phi_i$  for  $i = 1, 2, 3, 4$  and the albedo patterns  $S_a, S_b, S_c, S_d$  for  $i = 5$  and  $j = 1, 2, 3, 4$ , respectively. As such the likelihood relies on a difference between predicted and measured sensor responses.

The joint likelihood for the five data sets is found by taking the product of the likelihoods for each data set, since we expect that the standard deviations that were marginalized over to get the Student- $t$  distribution could have been different for each of the five data sets as they were not all recorded at the same time

$$\Pr(\mathbf{D} | \theta, I) = \prod_{i=1}^5 \Pr(D_i | \theta, I). \quad (18)$$

We employed nested sampling [15, 16] to explore the posterior probability since, in addition to providing parameter estimates, it is explicitly designed to perform evidence calculations, which we use to perform model comparison in identifying the most probable number of Gaussians in the MoG model. For each of the four MoG models (number of Gaussians varying from one to four) the nested sampling algorithm was initialized with 300 samples and iterated until the change in consecutive log-evidence values is less than  $10^{-8}$ . Typically one estimates the mean parameter values by taking an average of the samples weighted by a quantity computed by the nested sampling algorithm called the logWt in [15, 16]. Here we simply performed a logWt-weighted average of the sampled SSF fields computed using the sampled MoG model parameters (rather than the logWt-weighted average of the MoG model parameters themselves), so the result obtained using an MoG model consisting of a single Gaussian is not strictly a single two-dimensional Gaussian distribution. It is this discretized estimated SSF field matrix that is used directly in the convolution (9) to compute the likelihood functions as mentioned earlier in the last lines of Section 2.5.

### 3. Results and Discussion

In this section we present the SSF MoG light sensor model estimated from the laboratory data and evaluate its efficacy by demonstrating a significant improvement of the performance in the autonomous robotic platform.

*3.1. Light Sensor SSF MoG Model.* The light sensor data collected using the black-and-white boundary pattern are illustrated in Figure 5. One can see that the intensity recorded

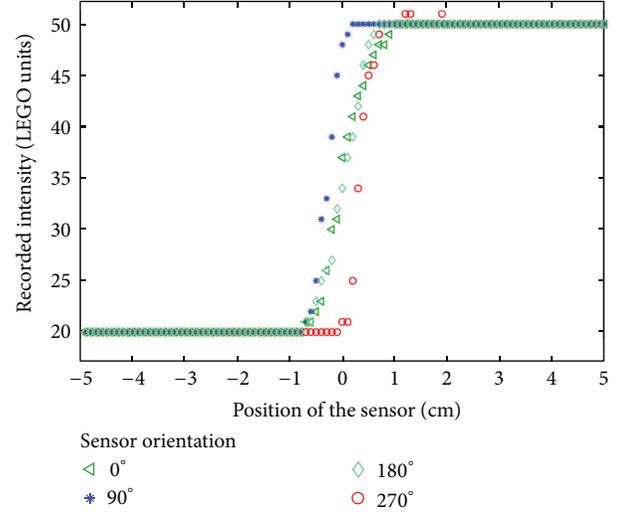


FIGURE 5: This figure illustrates the intensity measurements,  $D_1, D_2, D_3, D_4$  from the four sensor orientations, recorded from the sensor using the black-and-white boundary pattern. Figure 3 shows the orientations of the sensor corresponding to the symbols used in this figure.

TABLE 1: A comparison of the tested MoG SSF models and their respective log evidence (in units of  $\text{data}^{-408}$ ).

MoG model order	Log evidence	Number of parameters
1 Gaussian	$-665.5 \pm 0.3$	6
2 Gaussian	$-674.9 \pm 0.3$	12
3 Gaussian	$-671.9 \pm 0.4$	18
4 Gaussian	$-706.1 \pm 0.4$	24

by the sensor increases dramatically as the sensor crosses the boundary from the black region to the white region, but the change is not a step function, which indicates the finite size of the surface area integrated by the sensor. It is this effect that is to be modeled by the SSF function. There is an obvious asymmetry between the  $90^\circ$  and  $270^\circ$  orientations due to the shift of the transition region. In addition, there is a significant difference in slope of the transition region between the  $0^\circ, 180^\circ$  orientations and the  $90^\circ, 270^\circ$  orientations indicating a significant difference in the width of the SSF in those directions. Note also that the minimum recorded response is not zero, as the reflectance of the black surface is not completely zero.

The nested sampling algorithm produced the mean SSF fields for each of the MoG models tested, as well as the corresponding log evidence. Table 1, which lists the log evidence computed for each MoG model order, illustrates that the most probable model was obtained from the single two-dimensional Gaussian models by a factor of about  $\exp(9)$ , which means that it is about 8000 times more probable than the MoG consisting of two Gaussians. Figure 6 shows the mean SSF fields described by the MoG models of different orders. In all cases, the center of the SSF is shifted slightly above the physical center of the sensor package due to the placement of the photodiode (refer to Figure 1) as indicated by the data in Figure 5. In addition, as predicted, one sees

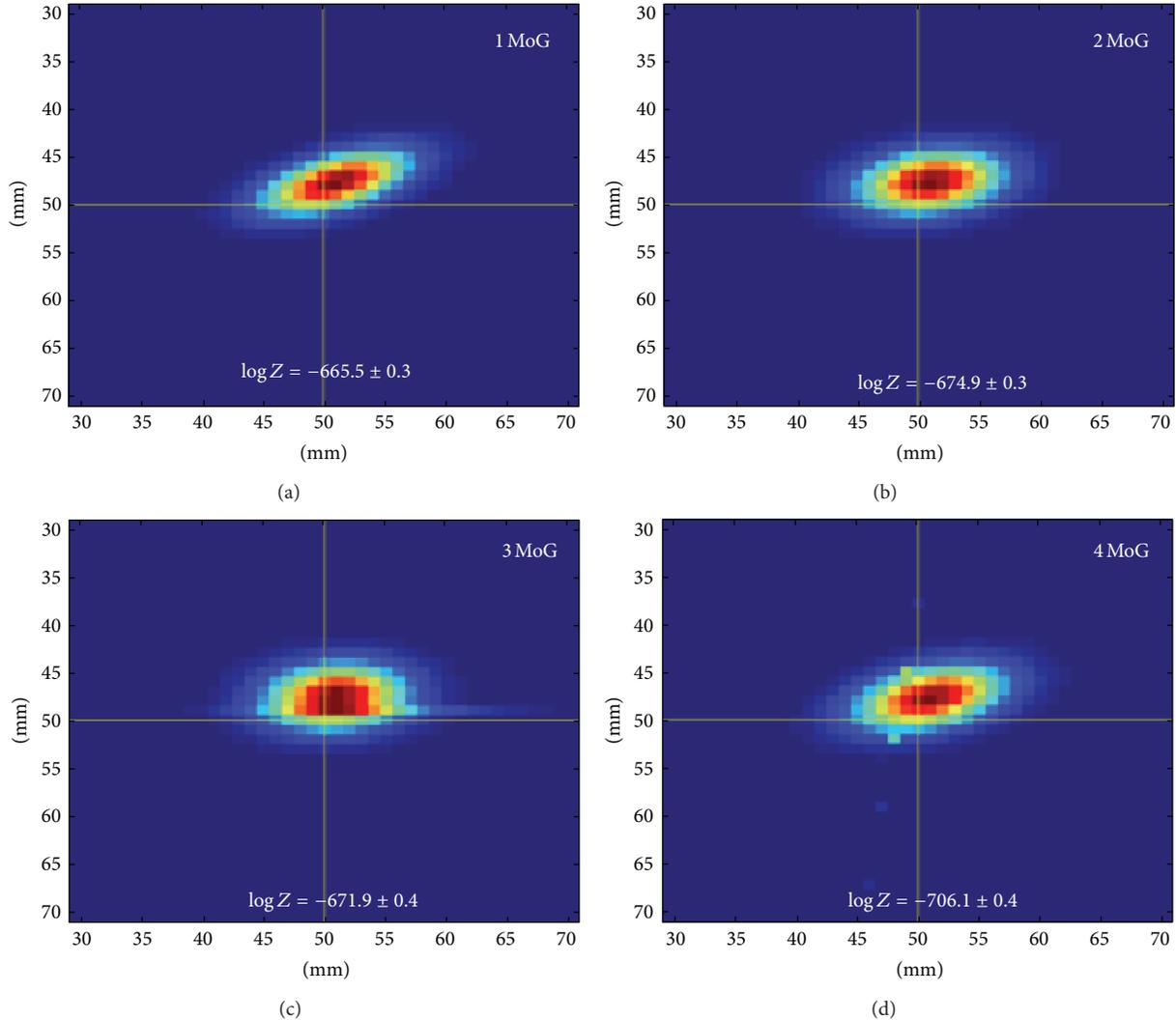


FIGURE 6: This figure illustrates the SSF obtained from the four MoG models along with their corresponding log-evidence values.

that the SSF is wider along the  $90^\circ$ – $270^\circ$  axis than along the  $0^\circ$ – $180^\circ$  axis. Last, Figure 7 demonstrates that the predicted light sensor output shows excellent similarity to the recorded data for the black-and-white boundary pattern.

In the next section, we demonstrate that explicit knowledge about how the light sensor integrates light arriving from within its field-of-view improves the inferences one can make from its output.

**3.2. Efficacy of Sensor Model.** The mean SSF field obtained using a single two-dimensional Gaussian model (Figure 6(a)) was incorporated into the likelihood function used by the robot’s machine learning system. Here we compare the robot’s performance in locating and characterizing a circle by observing the average number of measurements necessary to characterize the circle parameters within a precision of 4 mm (which is one-half of the spacing between the holes in the LEGO technic parts).

The three panels comprising Figure 8(a) illustrate the robot’s machine learning system’s view of the playing field using the naïve light sensor model. The previously obtained measurement locations used to obtain light sensor data are indicated by the black and white squares indicating the relative intensity with respect to the naïve light sensor model. The next selected measurement location is indicated by the green square. The blue circles represent the 50 hypothesized circles sampled from the posterior probability. The shaded background represents the entropy map, which indicates the measurement locations that promise to provide maximal information about the circle to be characterized. The low entropy area surrounding the white square indicates that the region is probably inside the white circle (not shown) and that measurements made there will not be as informative as measurements made elsewhere. Note that the circles partition the plane and that each partition has a uniform entropy. All measurement locations within that partition, or any other partition sharing the same entropy, all stand to be equally

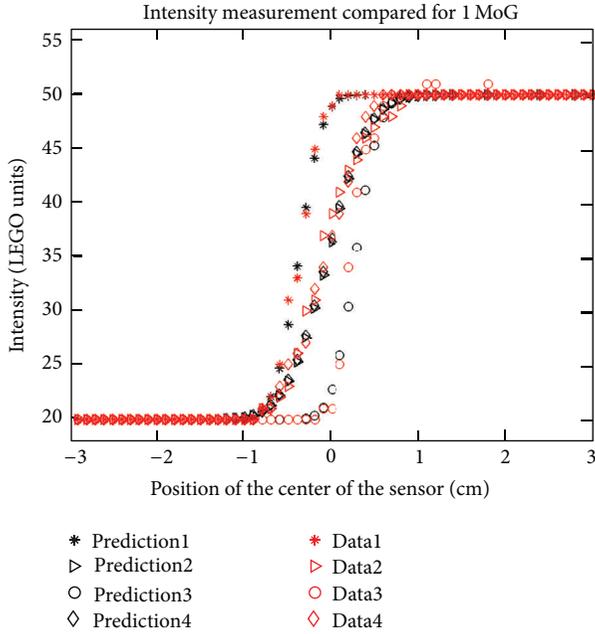


FIGURE 7: A comparison of the observed data (red) with predictions (black) made by the SSF field estimated using the single two-dimensional Gaussian MoG model.

informative. Here it is the fact that the shape is known to be a circle that is driving the likelihood function.

In contrast, the three panels comprising Figure 8(b) illustrate the playing field using the more accurate SSF model. Here one can see that the entropy is higher along the edges of the sampled circles. This indicates that the circle edges promise to provide more information than the centers of the partitioned regions. This is because that the SSF model enables one to detect not only whether the light sensor is situated above the circles edge but also how much of the SSF overlaps with the white circle. That is, it helps to identify not only whether the sensor is inside the circle (as is accomplished using the naïve light sensor model), but also the extent to which the sensor is on the edge of the circle. The additional information provided about the functioning of the light sensor translates directly into additional information about the albedo that results in the sensor output.

This additional information can be quantified by observing how many measurements the robot is required to take to obtain estimates of the circle parameters within the same precision in the cases of each light sensor model. Our experiments revealed that on average it takes  $19 \pm 1.8$  measurements using the naïve light sensor model compared to an average of  $11.6 \pm 3.9$  measurements for the more accurate SSF light sensor model.

#### 4. Conclusion

The quality of the inferences one makes from a sensor depend, not only on the quality of the data returned by the sensor, but also on the information one possesses about the sensor's

performance. In this paper we have demonstrated via a case study, how more precisely modeling a sensor's performance can improve the inferences one can make from its data. In this case, we demonstrated that one can achieve about 18% reduction in the number of measurements needed by a robot to make the same inferences by more precisely modeling its light sensor.

This paper demonstrates how a machine learning system that employs Bayesian inference (and inquiry) relies on the likelihood function of the data given the hypothesized model parameters. Rather than simply representing a noise model, the likelihood function quantifies the probability that a hypothesized situation could have given rise to the recorded data. By incorporating more information about the sensors (or equipment) used to record the data, one naturally is incorporating this information into the posterior probability, which results in one's inferences.

This is made even more apparent by a careful study of the experimental design problem that this particular robotic system is designed to explore. For example, it is easy to show that by using the naïve light sensor model, the entropy distribution for a proposed measurement location depends solely on the number of sampled circles for which the location is in the interior of the circle and the number of sampled circles for which the location is exterior to the circle. Given that we represented the posterior probability by sampling 50 circles, the maximum entropy occurs when the proposed measurement location is inside 25 circles (and outside 25 circles). As the robot's parameter estimates converge, one can show that the system is simply performing a binary search by asking "yes" or "no" questions, which implies that each measurement results in one bit of information. However, in the case where the robot employs an SSF model of the light sensor, the question the robot is essentially asking is more detailed: "to what degree does the circle overlap the light sensor's SSF?" The answer to such a question tends to provide more information, which significantly improves system performance. One can estimate the information gain achieved by employing the SSF model. Consider that the naïve model reveals that estimating the circle's position and radius with a precision of 4 mm given the prior information about the circle and the playing field requires 25 bits of information. The experiment using the SSF model requires on average 11.6 measurements, which implies that on average each measurement obtained using the SSF model provides about  $25/11.6 = 2.15$  bits of information. One must keep in mind, however, that this is due to the fact that the SSF model is being used not only to infer the circle parameters from the data, but also to select the measurement locations.

Because the method presented here is based on a very general inferential framework, these methods can easily be applied to other types of sensors and equipment in a wide variety of situations. If one has designed a robotic machine learning system to rely on likelihood functions, then sensor models can be incorporated in more or less a plug-and-play fashion. This not only promises to improve the quality of robotic systems forced to rely on lower quality sensors, but it also opens the possibility for calibration on the fly by updating sensor models as data are continually collected.

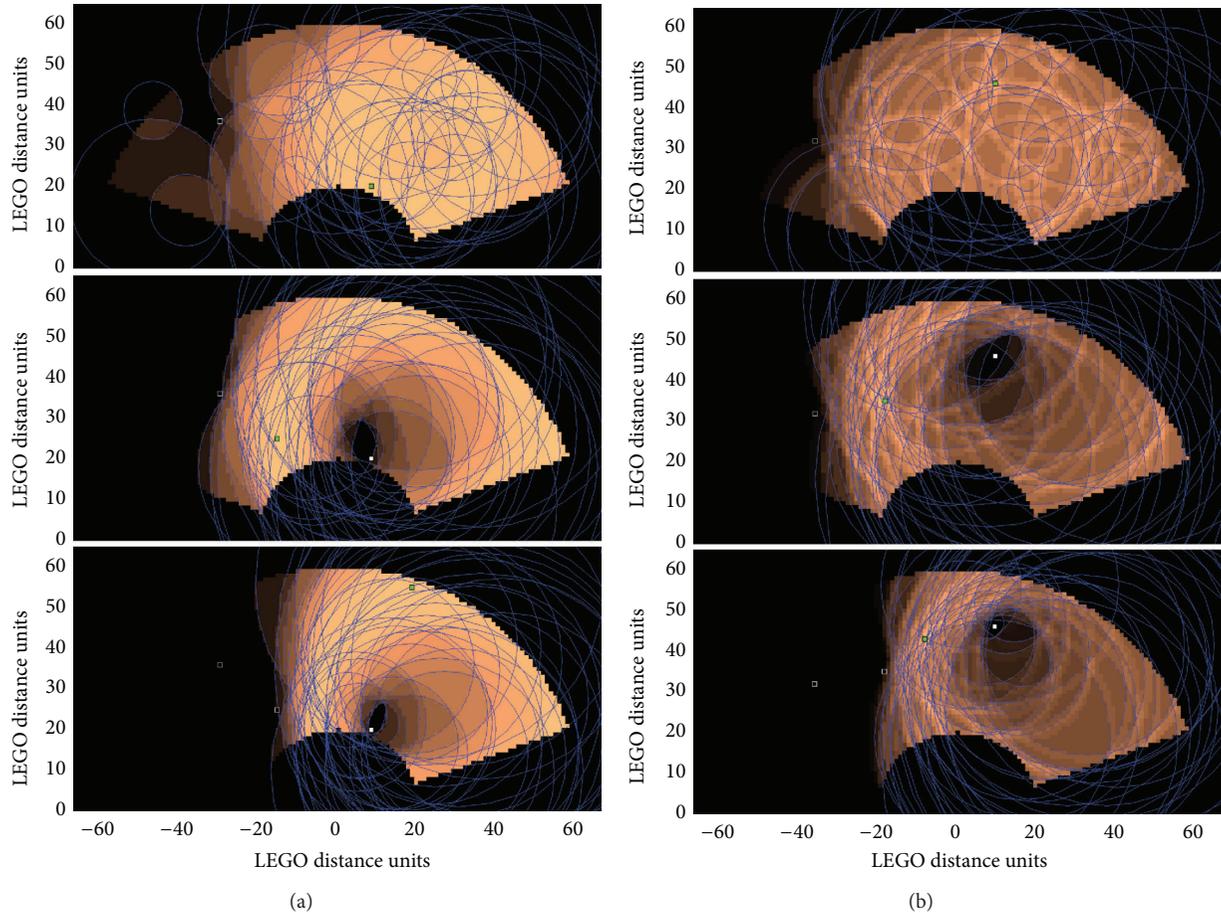


FIGURE 8: (a) These three panels illustrate the robot's machine learning system's view of the playing field using the naïve light sensor model as the system progresses through the first three measurements. The previously obtained measurement locations used to obtain light sensor data are indicated by the black-and-white squares indicating the relative intensity with respect to the naïve light sensor model. The next selected measurement location is indicated by the green square. The blue circles represent the 50 hypothesized circles sampled from the posterior probability. The shaded background represents the entropy map, which indicates the measurement locations that promise to provide maximal information about the circle to be characterized. Note that the low entropy area surrounding the white square indicates that the region is probably inside the white circle (not shown) and that measurements made there will not be as informative as measurements made elsewhere. The entropy map in Figure 2 shows the same experiment at a later stage after seven measurements have been recorded. (b) These three panels illustrate the robot's machine learning system's view of the playing field using the more accurate SSF light sensor model. Note that the entropy map reveals the circle edges to be highly informative. This is because it helps to identify not only whether the sensor is inside the circle (as is accomplished using the naïve light sensor model on the left), but also the extent to which the sensor is on the edge of the circle.

## Conflict of Interests

The authors have no conflict of interests to report.

## Acknowledgments

This research was supported in part by the University at Albany Faculty Research Awards Program (Knuth) and the University at Albany Benevolent Research Grant Award (Malakar). The authors would like to thank Scott Frasso and Rotem Guttman for their assistance in interfacing MatLab to the LEGO Brick. They would also like to thank Phil Erner and A. J. Mesiti for their preliminary efforts on the robotic arm experiments and sensor characterization.

## References

- [1] K. H. Knuth, P. M. Erner, and S. Frasso, "Designing intelligent instruments," in *Proceedings of the 27th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, K. Knuth, A. Caticha, J. Center, A. Giffin, and C. Rodriguez, Eds., vol. 954, pp. 203–211, AIP, July 2007.
- [2] K. H. Knuth and J. L. Center Jr., "Autonomous science platforms and question-asking machines," in *Proceedings of the 2nd International Workshop on Cognitive Information Processing (CIP '10)*, pp. 221–226, June 2010.
- [3] K. H. Knuth and J. L. Center, "Autonomous sensor placement," in *Proceedings of the IEEE International Conference on Technologies for Practical Robot Applications*, pp. 94–99, November 2008.

- [4] K. H. Knuth, "Intelligent machines in the twenty-first century: foundations of inference and inquiry," *Philosophical Transactions of the Royal Society A*, vol. 361, no. 1813, pp. 2859–2873, 2003.
- [5] N. K. Malakar and K. H. Knuth, "Entropy-based search algorithm for experimental design," in *Proceedings of the 30th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, A. M. Djafari, J. F. Bercher, and P. Bessière, Eds., vol. 1305, pp. 157–164, AIP, July 2010.
- [6] N. K. Malakar, K. H. Knuth, and D. J. Lary, "Maximum joint entropy and information-based collaboration of automated learning machines," in *Proceedings of the 31st International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, P. Goyal, A. Giffin, K. H. Knuth, and E. Vrscay, Eds., vol. 1443, pp. 230–237, AIP, 2012.
- [7] D. V. Lindley, "On a measure of the information provided by an experiment," *The Annals of Mathematical Statistics*, vol. 27, no. 4, pp. 986–1005, 1956.
- [8] V. V. Fedorov, *Theory of Optimal Experiments*, Probability and Mathematical Statistics, Academic Press, 1972.
- [9] K. Chaloner and I. Verdinelli, "Bayesian experimental design: a review," *Statistical Science*, vol. 10, pp. 273–304, 1995.
- [10] P. Sebastiani and H. P. Wynn, "Bayesian experimental design and shannon information," in *1997 Proceedings of the Section on Bayesian Statistical Science*, pp. 176–181, 1997, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.6037>.
- [11] P. Sebastiani and H. P. Wynn, "Maximum entropy sampling and optimal Bayesian experimental design," *Journal of the Royal Statistical Society B*, vol. 62, no. 1, pp. 145–157, 2000.
- [12] T. J. Loredo and D. F. Chernoff, "Bayesian adaptive exploration," in *Proceedings of the 23rd International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, G. Erickson and Y. Zhai, Eds., pp. 330–346, AIP, August 2003.
- [13] R. Fischer, "Bayesian experimental design—studies for fusion diagnostics," in *Proceedings of the 24th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, vol. 735, pp. 76–83, November 2004.
- [14] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, Chicago, Ill, USA, 1949.
- [15] J. Skilling, "Nested sampling," in *Proceedings of the 24th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, vol. 735, pp. 395–405, 2004.
- [16] D. Sivia and J. Skilling, *Data Analysis: A Bayesian Tutorial*, Oxford University Press, New York, NY, USA, 2nd edition, 2006.
- [17] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [18] W. K. Hastings, "Monte carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.
- [19] N. K. Malakar, A. J. Mesiti, and K. H. Knuth, "The spatial sensitivity function of a light sensor," in *Proceedings of the 29th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, P. Goggans and C. Y. Chan, Eds., vol. 1193 of *AIP Conference Proceedings*, pp. 352–359, American Institute of Physics, Oxford, Mass, USA, July 2009.

## Research Article

# Feasibility Investigation of Obstacle-Avoiding Sensors Unit without Image Processing

**Yasuhisa Omura, Hiroshi Masuda, and Yoshio Mimura**

*Graduate School of Science and Engineering, Kansai University, 3-3-35 Yamate-cho, Suita, Osaka 564-8680, Japan*

Correspondence should be addressed to Yasuhisa Omura; [omuray@kansai-u.ac.jp](mailto:omuray@kansai-u.ac.jp)

Received 11 February 2013; Accepted 9 June 2013

Academic Editor: Aiguo Song

Copyright © 2013 Yasuhisa Omura et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Feasibility of a simple method to detect step height, slope angle, and trench width using four infrared-light-source PSD range sensors is examined, and the reproducibility and accuracy of characteristic parameter detection are also examined. Detection error of upward slope angle is within 2.5 degrees, while it is shown that the detection error of downward slope angle exceeding 20 degrees is very large. In order to reduce such errors, a method to improve range-voltage performance of a range sensor is proposed, and its availability is demonstrated. We also show that increase in trial frequency is a better way, although so as not to increase the detection delay. Step height is identified with an error of  $\pm 1.5$  mm. It is shown that trench width cannot be reliably measured at this time. It is suggested that an additional method is needed if we have to advance the field of obstacle detection.

## 1. Introduction

In the last decade, autonomous mobile robots have been attracting wide attention, and technical levels have dramatically advanced (see, for instance, [1]). Many robots for entertainment, room cleaning, and other services have already been developed [2]. To be really practical, robots must be able to acquire environmental events and/or spatial information of their environment. Some robots for entertainment have optical sensors, ultrasonic sensors, touch sensors, and other configurations which have been implemented. To create more autonomous robots that suit future applications, the 2D infrared range sensor [3] and CMOS-imager camera [4] are being studied extensively. In these studies, sensor downsizing is an ongoing concern. However, the newly developed sensors are still expensive, and computing overhead is apt to increase. This is a fundamental problem with the present research roadmap.

2D path planning for mobile robots has also been studied extensively [5, 6]; it is considered that combining a path planning method [7, 8] with a potential-field method [9, 10] or a mapping technique is a promising approach. These techniques are also needed for future self-learning robots.

On the other hand, recently, a passive intelligent walker is proposed using a servo breaks [11]; in that trial, some obstacles (such as steep slope and steps) are detected. However, a user must change his/her front direction when the sensor has found an obstacle. In addition, the robot does not guide a better direction for walking to the user. Therefore, at least now, blind persons cannot use the walker.

In this paper, how to detect and classify obstacles in front of a robot without a camera [12–14] is investigated. The purpose of this paper is (1) to realize a sensor block that can detect the differences between step, slope, and trench, (2) to form arithmetic procedures to estimate characteristic values (step height, slope angle, and trench width), and (3) to propose algorithms that yield reliable judgments. Four infrared-light-source (IR) PSD range sensors are used. Experiments on the sensor block challenge its sensor functions with steps, slopes, and/or trenches.

The electrical or mechanical configuration of the testing robot is described in Section 2. Section 3 describes the measurement accuracy of the IR PSD range sensors used. Section 4 proposes algorithms that allow the robot to detect obstacles and estimate characteristic values. Section 5 describes the results of an obstacle-detection test and the

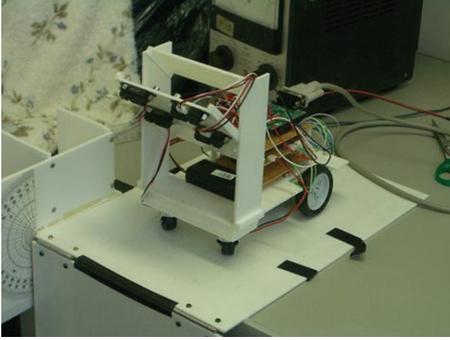


FIGURE 1: Photo of an assembled robot for testing.

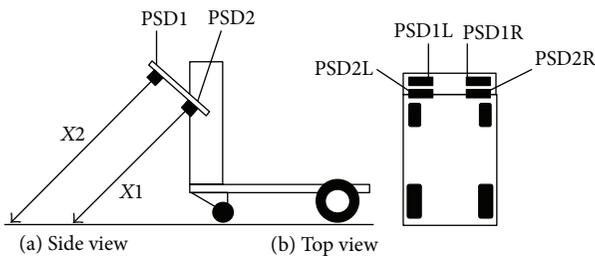


FIGURE 2: Schematic of sensors' layout.

reliability of obstacle recognition. Finally, the remaining issues are summarized.

## 2. Mechanical and Electrical Architecture of Testing Robot

A picture of the prototype robot to test sensor functions is shown in Figure 1. The testing robot has two nondriven caster wheels at the front and two motor-driven wheels at the back whose rotation speeds are controlled by a motor-drive circuit. The motor-driven wheels have four rotation modes (brake, stop, forward, and back). Since these four functions are implemented on the wheels independently, the robot can move in any direction. Four range sensors are placed on the front of the testing robot (PSD1L, PSD1R, PSD2L, and PSD2R, resp.) to detect obstacles in front of the testing robot (see Figure 2). These four sensors detect distances from the sensor to the floor, and the microcontroller calculates characteristic values, for example, the slope angle  $\theta$  when the obstacle is a slope.

The electronic architecture of the testing robot is shown in Figure 3. The circuit-mounted board includes a microcontroller (ADuC7026 [15] produced by Analog Devices Corp.) to give the robot a data processing function. The microcontroller has input terminals for up to 12 single-ended A/D converters and other analog processing functions. The microcontroller receives analog signals from sensors through its built-in A/D converters, logically assigns the environment to one of the obstacles or no obstacle, and finally outputs the characteristic value of the obstacle (slope angle  $\theta$  for the slope, step height  $h_1$  for the step, and so on). The microcontroller

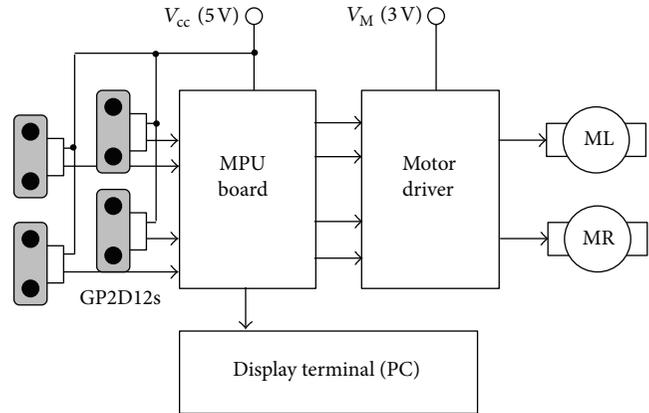


FIGURE 3: Electronic control system for the motor driver and others.

on the MPU board calculates the obstacle's dimensions and transfers the data to a PC via the RC-232 interface.

## 3. Accuracy and Reproducibility of Output Signal of PSD Sensor

The first step is to evaluate the potential of the IR range sensor (GP2D12 [16] produced by SHARP Corp.) used to detect obstacles; we focus here on the sensor performance attributes not described in the commercial data sheet. This sensor unit has the following features.

- (1) Distance detection range (sensor to object) is 10 to 80 cm in the present case. When GP2Y0A02YK is used, however, the distance detection range is 20 to 150 cm. In this experiment, we employed GP2D12 because of easy verification of proposal.
- (2) IR source signal of one sensor interferes very little with the functioning of the other sensors.
- (3) The sensor is basically insensitive to object color and reflectivity.
- (4) The sensor is basically insensitive to room light.
- (5) Distance from the sensor to the floor can be detected even when the object surface is tilted. However, the variation in range is significant when the tilt angle is large.
- (6) Low cost and small size.

As just described, the IR PSD sensor has many advantages over other sensors. In some cases, however, there is a significant amount of electrical noise in the output signal when we consider some applications that demand the detection of slope angle. This suggests that how accurately the sensor detects distance ( $X_1$ ,  $X_2$ ) before an accurate sensor circuit block is designed has to be examined.

As an example, range data created by transforming the analog signals of the IR PSD sensors are shown in Figures 4 and 5; Figure 4 shows the output of the microcontroller when challenged with an 18 mm high upward step, and Figure 5 shows that for a 20-degree downward slope. In both cases, the

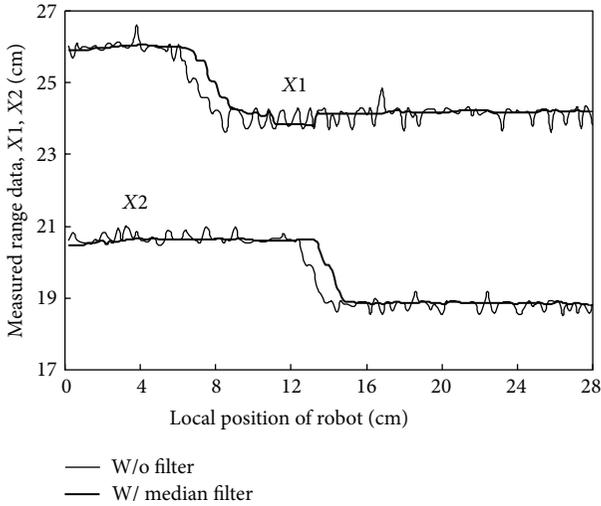


FIGURE 4: Range data evolution when the robot is approaching an upward step.

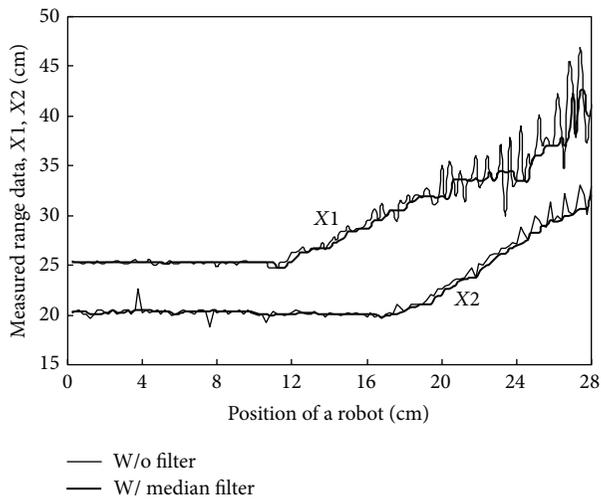


FIGURE 5: Range data evolution when the robot is approaching a downward slope.

testing robot had a constant velocity on the floor. In Figures 4 and 5, the thin lines are the unprocessed digital range data transferred from the microcontroller, while the bold lines are the range data after being passed through a median filter (window number  $N1 = 5$ ) (see the appendix).

Figure 4 shows that the median filter is effective in removing the impulse noise. It also shows that the filter yields a time delay, resulting in a 5 mm local position difference in the case of  $N1 = 5$ . The noise can be further reduced by increasing  $N1$ , but at the cost of simultaneously increasing the time delay. Because of this tradeoff, it is preferable to adjust  $N1$  to suit the application.

In Figure 5, the impulse noise is sufficiently removed at short distances as well as in Figure 4, but not at distances beyond 50 cm. This is due to the sensor's performance limitation [16]; when  $X1 > 50$  cm, even a small voltage shift

of output signal of sensor results in a large variation in range data. When the angle between the IR-light beam from the sensor and the object surface increases, the IR signal returned attenuates, and the influence of room light becomes significant. This means that a downward slope yields a large variation in the detected signal.

#### 4. Method of Extracting Spatial Values

In Section 4.1, how the sensing circuit block identifies steps, slopes, and trenches using the upper and lower sensors (PSD1 and PSD2) is described. Section 4.2 describes the mathematical model that the sensing unit applies to the calculation of step height, slope angle, or trench width. Section 4.3 details the results of experiments on the determination of step height, slope angle, and trench width.

In this chapter, it is assumed that the testing robot directly faces the obstacle (the width of which is taken to be effectively infinite). Note that all the range data ( $X1L$ ,  $X1R$ ,  $X2L$ , and  $X2R$ ) displayed in the figures are the result of median filtering. Results obtained assuming more practical situations are shown in Section 5.

*4.1. How to Classify Slopes, Steps, and Trenches.* First, the notations used in this section are explained.  $X1$  and  $X2$  stand for the distances given by PSD1 and PSD2, respectively. When the testing robot runs on a flat floor, it is assumed that PSD1 and PSD2 yield distance data  $X1o$  and  $X2o$ , respectively. In a practical situation, various noises in the data yielded by the sensors should be taken account of. Accordingly, we introduce positive threshold values of  $X1T$  and  $X2T$  to improve the detection reproducibility of distance data when determining whether the event (i.e., slope, step, or trench) has occurred. When PSD1 outputs data satisfying the condition of  $|X1 - X1o| < X1T$ , the testing robot “thinks” that it is on a flat floor. In this case, we say that  $S(\text{PSD1}) = \text{“Flat”}$ . When PSD1 outputs data satisfying the condition of  $|X1 - X1o| > X1T$ , the sensing circuit block “thinks” that it may be facing a slope, a step, or a trench. In this case, we say that  $S(\text{PSD1}) = \text{“NON-F”}$ . In the present experiment, we empirically set  $X1T = 0.8$  [cm] and  $X2T = 0.5$  [cm] by taking account of the noise level shown in Figures 4 and 5, respectively. For example, the testing robot is running on a flat floor, when the “states” output by the 4 sensors are “flat”, and we use the following descriptions:

$$\begin{aligned}
 S(\text{PSD1L}) &= \text{“Flat”}, \\
 S(\text{PSD1R}) &= \text{“Flat”}, \\
 S(\text{PSD2L}) &= \text{“Flat”}, \\
 S(\text{PSD2R}) &= \text{“Flat”}.
 \end{aligned} \tag{1}$$

Next, how the sensing circuit block uses the trigonometric method shown in Figure 6 in order to extract geometrical

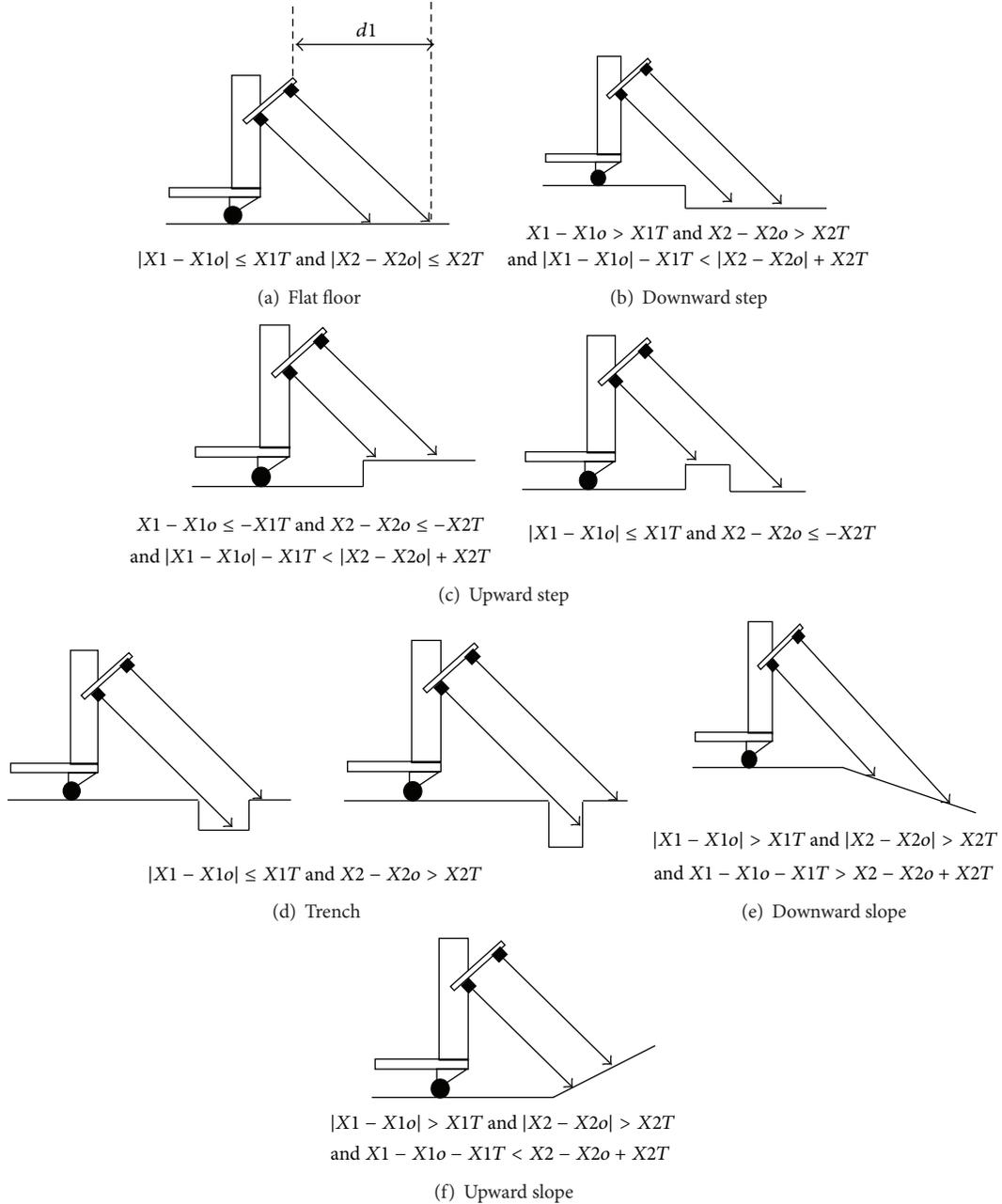


FIGURE 6: How to classify slopes, steps, and trenches. Mathematical algorithms are shown.

parameters of different slopes, steps, and trenches from data obtained is described.

(a) *Flat Floor*. When the sensing circuit block compares the range data to the threshold value given in the previous section, and PSD1 and PSD2 output data satisfying the condition of  $S(\text{PSD1}) = \text{"FLAT"}$  &  $S(\text{PSD2}) = \text{"FLAT"}$ , the sensing circuit block "thinks" that it is on a flat floor (see Figure 6(a)). The equivalent mathematical relationship can be expressed as

$$|X1 - X1o| \leq X1T, \quad |X2 - X2o| \leq X2T. \quad (2)$$

(b) *Downward Step*. When PSD1 and PSD2 output data satisfying the following condition, the sensing circuit block "thinks" that it is facing a downward step (see Figure 6(b)):

$$\begin{aligned} X1 - X1o > X1T, \quad X2 - X2o > X2T, \\ |X1 - X1o| - X1T < |X2 - X2o| + X2T. \end{aligned} \quad (3)$$

(c) *Upward Step*. When PSD1 and PSD2 output data satisfying the following condition, the sensing circuit block "thinks" that it is facing an upward step (see Figure 6(c)):

$$|X1 - X1o| \leq X1T, \quad X2 - X2o \leq -X2T. \quad (4)$$

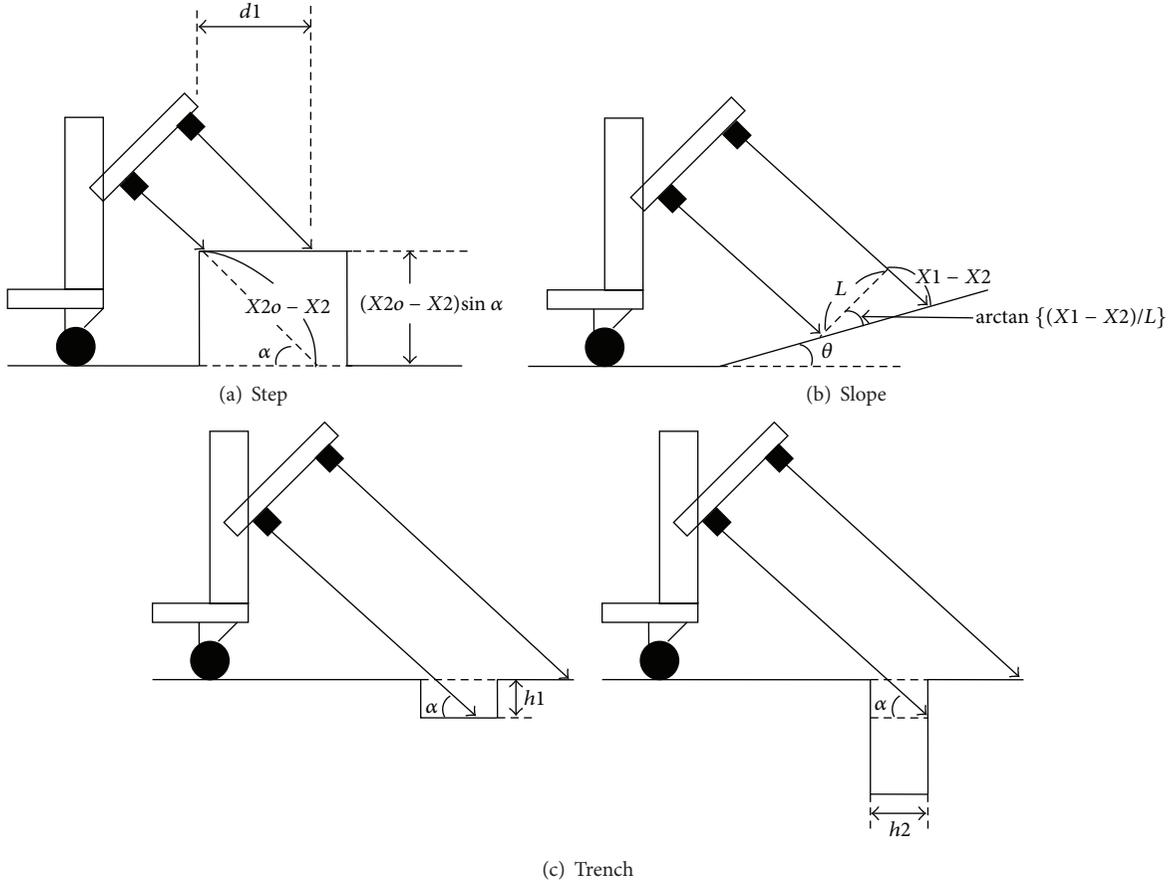


FIGURE 7: How to classify slopes, steps, and trenches. Characteristic parameter extraction method is shown.

(d) *Trench*. When PSD1 and PSD2 output data satisfying the following condition, the sensing circuit block “thinks” that it is facing a trench (see Figure 6(d)). In this case, the PSD sensor receives the IR signal reflected from the “side wall of the trench” or the IR signal reflected from the “bottom of the trench” as shown in Figure 7(c). The algorithm in the following form cannot distinguish these two cases:

$$|X1 - X1o| \leq X1T, \quad X2 - X2o > X2T. \quad (5)$$

Details are discussed later.

(e) *Downward Slope*. When PSD1 and PSD2 output data satisfying the following condition, the sensing circuit block “thinks” that it is facing a down slope (see Figure 6(e)):

$$|X1 - X1o| > X1T, \quad |X2 - X2o| > X2T, \\ X1 - X1o - X1T > X2 - X2o + X2T. \quad (6)$$

(f) *Upward Slope*. When PSD1 and PSD2 output data satisfying the following condition, the sensing circuit block “thinks” that it is facing an upward slope (see Figure 6(f)):

$$|X1 - X1o| > X1T, \quad |X2 - X2o| > X2T, \\ X1 - X1o - X1T < X2 - X2o + X2T. \quad (7)$$

4.2. *Equations to Calculate Slope Angle, Step Height, and Trench Width*. Slope angle, step height, and trench width can be calculated from range data (X1, X2). Figure 7 visualizes the trigonometric techniques used.

(a) *Step Height*. Step height  $h1$  is calculated using (8). A schematic is shown in Figure 7(a):

$$h1 = (X2o - X2) \sin \alpha, \quad (8)$$

where  $\alpha$  is the angle of sensor signal against a flat floor (here,  $\alpha = 45^\circ$ ). When  $h1 < 0$ , a downward step is suggested, and when  $h1 > 0$ , an upward step is suggested.

(b) *Slope Angle*. Slope angle  $\theta$  is calculated using (9). A schematic is shown in Figure 7(b):

$$\theta = \frac{\pi}{2} - \alpha - \arctan\left(\frac{(X1 - X2)}{L}\right). \quad (9)$$

When  $\theta < 0$ , a downward slope is suggested, and when  $\theta > 0$ , an upward slope is suggested.

(c) *Trench Width.* When the robot faces a trench, the sensor output differs between two cases: (i) the sensor signal is reflected from the bottom of the trench and (ii) the sensor signal is reflected from a side wall of the trench. Initially, the sensing circuit block cannot judge which is correct. A possible solution is to force the sensing circuit block to calculate two dimensions, trench depth and trench width. Trench depth  $h1$  is calculated using the following equation:

$$h1 = (X2o - X2) \sin \alpha. \quad (10)$$

Trench width  $h2$  is calculated by the next equation:

$$h2 = \max \{(X2o - X2) \cos \alpha\}, \quad (11)$$

where  $\max\{X\}$  returns the maximal value of  $X$ .

Since all sensors are positioned so that their surfaces are angled at  $45^\circ$  against a flat floor, calculated values of  $h1$  and  $h2$  are identical. When the robot approaches the trench, the judgment of whether it can cross the trench depends on the diameter ( $D_w$ ) of wheels of testing robot. Consider case (i). When  $D_w$  is much larger than the trench depth, the robot may be able to cross the trench. Consider case (ii). When  $D_w$  is much larger than the trench width, the robot can go over the trench. Therefore, the robot can pass through the trench for both cases, (i) and (ii), when  $h1$  or  $h2$  is much smaller than  $D_w$ . In other words, it is not necessary for us to distinguish cases (i) and (ii); we can apply (10) to decide whether the testing robot can go forward or not when the robot detects a trench.

In practical applications, the sensors do not always yield precise characteristic values to use the above equations because of various noises (including external disturbance) or spatial dispersion of the emitted IR signal. This suggests the need for some additional method to guarantee the accuracy or the reproducibility of the characteristic values and judgment reliability; detail is given in Section 5.

*4.3. Measurement Results: Step Height, Slope Angle, and Trench Width.* Measurement results of a step height for which the testing robot should stop in front of a step are summarized in Table 1; 1000 sensing trials were averaged in each event of obstacle discovery, and the medial filter number ( $N1$ ) was 5. As is evident in Table 1, the variation of evaluated step height  $h1$  is very small; the difference between the maximal value and the minimal value is about 3 mm for the upward step and about 6 mm for the downward step. We can see that the present evaluation technique does not always yield accurate data.

Measurement results of upward slope angle ( $\theta$ ) are shown in Table 2; 1000 sensing trials were averaged in each event of obstacles, and the medial filter number was 5 or 21. In the experiment, we assumed 3 cases for the horizontal distance ( $d1$ ) between the front edge of the testing robot and the boundary of the flat floor and the slope, that is, 0 cm, 4 cm, and 7 cm. The slope angles were  $20^\circ$ ,  $15^\circ$ , and  $10^\circ$ . It can be seen in Table 2 that the averaged value of  $\theta$  basically increases

TABLE 1: Step height evaluation results in units of cm. 1000 sensing trials are averaged. The medial filter number ( $N1$ ) is 5. The real step height ( $h1$ ) is 1.30 cm.

	Upward step height (cm)	Downward step height (cm)
Mean value	1.30	-1.33
Max. value	1.45	-1.00
Min. value	1.18	-1.56
Variance	0.0057	0.0019

with  $d1$ . In this study, the slope angle evaluation algorithm does not estimate distance  $d1$ , and so the robot estimates characteristic values without stopping as it approaches the obstacle, resulting in a slight drop in accuracy. It is also seen in Table 2 that a large  $N1$  value reduces the variation in estimated values, although many trials of measurement waste time before judgment. In addition, for  $N1 = 21$ , the difference between the maximal value and the minimal value is not always reduced.

Table 3 shows measurement results of downward slope angle. As is evident in Table 3, the variation of measurement results is very large in contrast to the upward slope values. This suggests the need to improve judgment reliability for practical applications.

## 5. Dynamic Detection of Obstacles in a Test Road

In Sections 5.1 to 5.4, we describe an algorithm to be used in practical situations. Section 5.5 details the accuracy of several evaluations.

*5.1. Logical Flow.* A schematic flow showing how the testing robot avoids obstacles is shown in Figure 8. First, when the left or the right sensor state is “NON-F”, the testing robot changes its position so that the front line of the testing robot keeps being parallel to the border line of the obstacle and the flat floor. Next, the testing robot approaches the border line, again detects signals from the obstacle, and subsequently concludes whether the obstacle facing it is a slope, step, or trench. Finally, when the testing robot recognizes that the obstacle is a step, it calculates the tentative step height, compares the calculated value to the threshold value, and then concludes whether it has to avoid the obstacle or not. When the testing robot detects a slope or a trench, the testing robot traces the same logical flow. As just described, in order to successfully classify the obstacle and to get reliable characteristic values, causes of the errors in detecting the signals from the obstacle must be analyzed.

*5.2. Aligning the Testing Robot to the Obstacle.* In this section, we describe how the testing robot positions itself in the vicinity of the obstacle. For all obstacles, the testing robot should directly face the obstacle to maximize the detection accuracy; this is the most important point in detecting the

TABLE 2: Upward slope angle evaluation results in units of cm. 1000 sensing trials are averaged. The medial filter number ( $N1$ ) is 5 or 21.

Medial filter number ( $N1$ )	Real slope angle (deg.)	$d1$ (cm)	Mean value (deg.)	Max. value (deg.)	Min. value (deg.)	Variance (deg.)
5	20.0	0.00	17.0	19.4	15.9	0.465
		4.00	18.9	21.0	17.5	0.561
		7.00	20.5	22.2	17.4	1.01
	15.0	0.00	14.7	16.5	13.0	0.218
		4.00	15.3	16.2	14.2	0.121
		7.00	16.1	18.7	12.8	1.89
	10.0	0.00	9.68	12.1	7.74	0.243
		4.00	10.3	13.4	7.84	1.36
		7.00	10.8	13.2	9.26	0.162
21	20.0	0.00	16.9	18.1	16.5	0.0951
		4.00	19.7	20.5	18.9	0.110
		7.00	19.3	22.2	18.4	0.284
	15.0	0.00	14.6	15.3	13.0	0.104
		4.00	14.8	17.6	13.5	0.873
		7.00	15.5	16.3	14.8	0.129
	10.0	0.00	9.80	10.5	9.11	0.0630
		4.00	9.89	10.7	8.21	0.114
		7.00	9.40	11.5	8.59	0.255

TABLE 3: Downward slope angle evaluation results in units of cm. 1000 sensing trials are averaged. The medial filter number ( $N1$ ) is 21.

Real slope angle (deg.)	$d1$ (cm)	Mean value (deg.)	Max. value (deg.)	Min. value (deg.)	Variance (deg.)
-20.0	10.0	-19.9	-13.7	-23.2	2.06
	5.00	-18.9	-10.4	-25.4	7.69
	2.00	-18.4	-10.4	-27.2	10.3
-15.0	10.0	-14.6	-10.1	-18.0	1.41
	5.00	-14.2	-10.0	-21.7	3.76
	2.00	-14.5	-10.0	-19.8	4.35
-10.0	10.0	-10.6	-7.55	-13.7	1.45
	5.00	NA	NA	NA	NA
	2.00	NA	NA	NA	NA

parameters of an obstacle. This process is detailed below (see Figure 9).

- (1) First, the testing robot approaches the obstacle, receives range data, and examines whether the data satisfies the condition  $S(\text{PSD2L}) = \text{"NON-F"} \wedge S(\text{PSD2R}) = \text{"FLAT"}$  (see Figure 9(a)).
- (2) The testing robot moves forward slightly, again receives range data, and examines whether the data satisfies the condition  $S(\text{PSD2L}) = \text{"NON-F"} \wedge S(\text{PSD2R}) = \text{"NON-F"}$  (see Figure 9(b)).
- (3) When both PSD2L and PSD2R detect the "NON-F" signals, the testing robot moves as follows (see Figure 9(c)).
  - (i) When  $S(\text{PSD2L}) = \text{"NON-F"}$ , for example, the left motor reverses.

- (ii) When  $S(\text{PSD2L}) = \text{"FLAT"}$ , the left motor idles.
- (iii) When  $S(\text{PSD2R}) = \text{"NON-F"}$ , the right motor reverses.
- (iv) When  $S(\text{PSD2R}) = \text{"FLAT"}$ , the right motor idles.

The above algorithm ensures that the testing robot directly faces the obstacle.

- (4) When both motors stop, the algorithm has successfully terminated, and the testing robot approaches the obstacle again.

In the detecting process from (1) to (2), when the time interval of "NON-F" events of two sensors is longer than a certain value, the testing robot turns around before reaching the expected obstacle. In other words, the testing robot does

TABLE 4: Successful trials in detecting slopes. In the detection of the slope angle, detected data are averaged with 100 trials.

Real slope angle (deg.)	Successful detection rate	Mean value (deg.)	Max. value (deg.)	Min. value (deg.)	Variance (deg.)
-20.0	100/100	-20.0	-18.3	-24.1	0.926
-10.0	100/100	-12.7	-11.4	-14.0	0.261
10.0	100/100	10.3	10.9	9.52	0.0748
20.0	100/100	21.4	22.3	20.1	0.168

not estimate the vertical offset when the incident angle is very small.

The present algorithm yields a small degree of uncertainty on the testing robot's alignment due to the use of the threshold values  $X1T$  and  $X2T$ . In this experiment, we found an alignment error of up to  $10^\circ$ . Later, we evaluate the influence of this alignment error on the determination of characteristic parameters.

**5.3. How to Classify Obstacles Using Sensor Pairs.** First, we explain how to classify slopes, steps, and trenches (see Figure 8). Using the simple method described in Section 4, the testing robot may, for example, incorrectly classify a real slope as a step or a flat floor. This erroneous judgment comes from sensor noise and relatively large threshold values ( $X2T$  and  $X1T$ ). When these threshold values are large, the erroneous judgment becomes more common. To avoid this difficulty, we force the testing robot to calculate characteristic values repeatedly and to get the mean or median value. This flow is described below.

- (1) When the robot detects an object, it calculates the characteristic values 20 times and stores these data in memory.
- (2) The robot classifies the obstacle according to the highest frequency of classification after the 20 trials.
- (3) When the frequency of "trench" exceeds 5 in the 20 trials, the testing robot classifies the obstacle as a trench.

We have confirmed that this majority-decision process reduces the frequency of erroneous judgment.

**5.4. How to Calculate the Characteristic Values of a Specific Obstacle.** Here we describe a method for calculating the characteristic values.

- (1) When upward (or downward) step height  $h1$  is calculated, the testing robot calculates the mean of 20 trials.
- (2) When upward (or downward) slope angle  $\theta$  is calculated, the testing robot calculates the mean of 20 trials.

Next, when the testing robot stops, the testing robot gets the data set of  $X1L$ ,  $X1R$ ,  $X2L$ , and  $X2R$  200 times. After determining the mean values of  $X1L$ ,  $X1R$ ,  $X2L$ , and  $X2R$ , they are labeled  $X1L'$ ,  $X1R'$ ,  $X2L'$ , and  $X2R'$ , respectively. Finally, using values of  $X1L'$ ,  $X1R'$ ,  $X2L'$ , and  $X2R'$  and (9), the testing robot calculates the downward slope angle  $\theta$ . This technique is very powerful in suppressing noise (as described

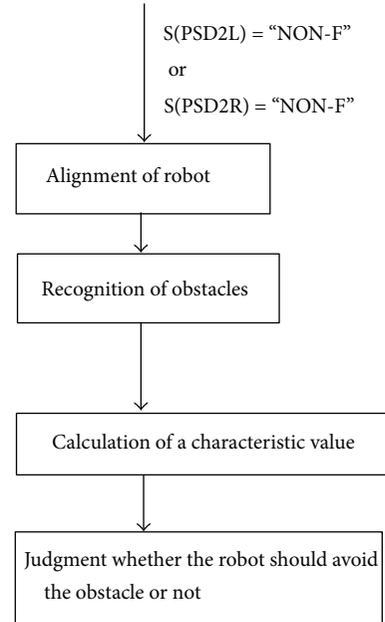


FIGURE 8: Algorithm for the robot to detect an obstacle and to avoid it.

in Sections 3 and 4). This benefit incurs the cost a 5 sec. delay in determining the downward slope angle. This limitation stems from the delay time of the PSD range sensor used in this study. We must employ a fast-processing PSD range sensor in the future.

**5.5. Evaluation Results of Characteristic Values.** Tables 4 to 6 show the characteristic values yielded by the logical process described in the previous section. Table 4 shows slope angle values extracted from signals given by sensors mounted on the testing robot; the offset value of sensor signals is considered in calculating the characteristic values, and the testing robot logically determines which obstacle has been encountered. As a result, the testing robot showed a very few errors in the classification of obstacles. Erroneous judgment, however, sometimes takes place in case of a gentle slope, which depends on threshold values of  $X1T$  and  $X2T$ . The gentle slope sometimes gives the sensor a noisy signal that cannot be easily detected as meaningful data; in this case, the testing robot fails to correctly determine the slope angle. Raising the values of  $X1T$  and  $X2T$  yields more conclusive data by sacrificing the detectable range of slope angle.

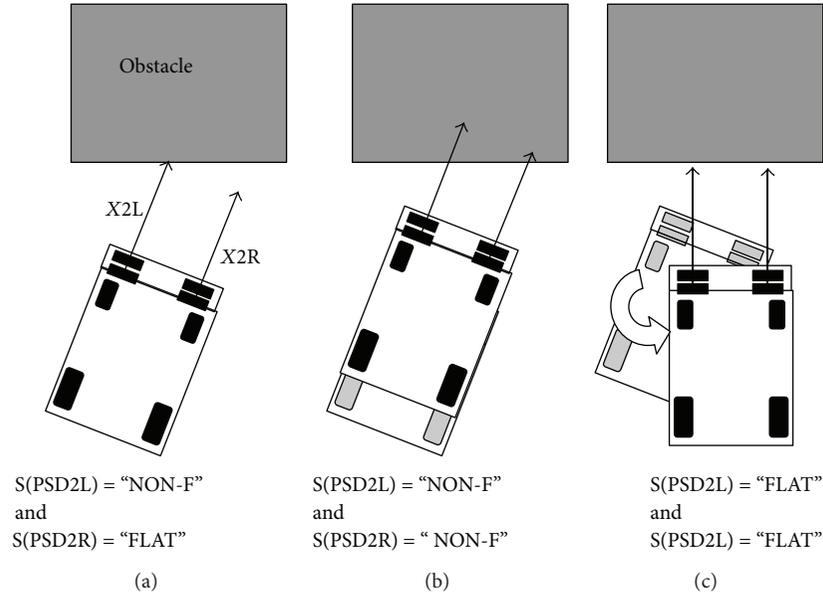


FIGURE 9: Front line alignment of the robot facing the obstacle.

Next, we discuss the accuracy of extracted slope angle  $\theta$ . In the present experiment, the detectable range of slope angle  $\theta$  is  $20^\circ$  to  $-10^\circ$ , and the deviation of extracted slope angle is at most  $2.5^\circ$ . At  $\theta = -20^\circ$ , however, the uncertainty in detected angle rises to  $4^\circ$ . When the angle of the sensor-light incident on the object's surface becomes small, the intensity of reflected-light signal becomes very weak; this results in a lower dynamic range in the sensor's output signal. This is basically the same phenomenon described in Section 3.

Since the sensor emits an infrared light signal, the reflection rate of the light depends on the color of the object's surface. In addition, the sensor's output attenuates as the distance of the sensor from the object increases. When the color of the object is dark, the sensor's output falls as does its dynamic range. As a result, it is usually difficult to accurately detect the angle of a steep slope. One possible way to remove this difficulty is to use a PSD sensor whose distance-output-voltage characteristic is almost linear or to widen the window of the median filter, although this would increase the detection time.

Table 5 shows step height values that are recalculated by the sensor module with some offset angle when the testing robot approaches the step. Erroneous detection did not occur. The detected step height was more accurate than that described in Section 4, where the signal-filtering technique was applied to the step-height detection. When the maximal step height that forces the robot to back away is 13 mm, the maximal value of calculated step height should be 11 mm because the maximal variation in the output voltage signal of the sensor is equivalent to the step height of 2 mm.

Table 6 shows trench-width values determined by the testing robot; the offset value of sensor signals was considered in calculating the characteristic values.

The basic algorithm used to detect a trench was described in Section 3. When the testing robot approaches the trench

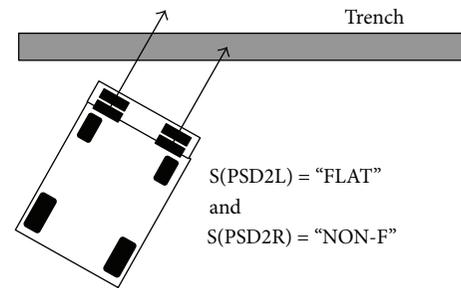


FIGURE 10: How to detect a trench.

at an oblique angle (see Figure 10), the correction of trench depth, used in the algorithm described in Section 5.1, cannot be employed because the algorithm assumes a direct approach to the trench. In this experiment, the testing robot was limited to approaching the trench at nearly  $90^\circ$ . This experiment was made on two trenches with different sizes.

As is seen in Table 6, when the trench width is reduced, the frequency of erroneous judgment rises. One cause is the incompleteness of the detection algorithm; the testing robot incorrectly judges the trench as a flat floor. One way of overcoming this difficulty is to reduce the values of  $X1T$  and  $X2T$ . However, it raises the remaining electrical noise in the output signal. Another approach is to increase the diameter of the wheels of testing robot.

**5.6. Advantage of the Method Proposed.** Recently many mathematical techniques are proposed for the purpose of monitoring the changing environment [17], multirobot navigation [18], motion tracking [19], self-collision avoidance [20], blind juggler control [21], precise positioning [22], distant control [23], and motion grammar description [24].

TABLE 5: Successful trials in detecting steps. In the detection of the step height, detected data are averaged with 100 trials.

Real step height (cm)	Successful detection rate	Mean value (cm)	Max. value (cm)	Min. value (cm)	Variance (cm)
1.30	100/100	1.37	1.47	1.17	0.0017
-1.30	100/100	-1.21	-1.11	-1.33	0.0022

TABLE 6: Successful trials in detecting trenches. In the detection of the trench width, detected data are averaged with 100 trials.

Real trench dimension	Successful detection rate	Mean value (cm)	Max. value (cm)	Min. value (cm)	Variance (cm)
1.30 (cm) $\times$ 1.30 (cm)	55/100	-0.94	-0.85	-1.26	0.0098
2.00 (cm) $\times$ 2.00 (cm)	90/100	-1.67	-1.24	-2.32	0.0310

In most proposals, the algorithm is complex [17–20], and/or the time derivatives are applied to the parameter analysis [20, 21]. The time derivative frequently yields extra noise in the signal analysis [21]. On the other hand, the use of many controllable degrees of freedom (DOF) [19, 20, 22] leads us to a large RMS of errors. Polynomial formalism for the position control [23] requests many topological definitions to realize reliable forward kinematics. Application of motion grammar to robots [24] also requests many possible logical patterns to avoid undesirable actions.

As is suggested in the previous articles, complex mechanics and complex actions request complex algorithms, resulting in a high cost and much difficulty. We think that the technique applied to some robots (cleaning robot, visitor-guide robot, and so on) requests simplicity of electronics and software from the cost of product. Therefore, the method proposed here has an advantage from the point of view of system volume and its cost.

## 6. Concluding Remarks

We have proposed a simple method for detecting the step height, slope angle, and trench width using four PSD range sensors (GP2D12) and have examined the reproducibility and accuracy of characteristic parameter detection. Detection error of upward slope angle is about  $2.5^\circ$ , while the detection error for downward slope angles exceeding  $20^\circ$  is very large. To reduce these errors, we have to use a range sensor that offers better range-voltage performance, or we have to increase the trial frequency so as not to increase the detection delay. Step height is extracted with an error of  $\pm 1.5$  mm. The current algorithm for trench width is not so accurate. It is suggested that an additional method must be introduced to advance the obstacle detection technique. However, this study has demonstrated that obstacle detection is basically possible without image processing.

## Appendix

### Median Filter Algorithm

In this paper, we used the following algorithm in order to reduce the electrical noise in the original signal. First, we get

$N$  datum points from the microcontroller. After sorting the  $N$  data ( $D[1]$  to  $D[N]$ ), we extract the maximal value,  $D_{\max}$ , and the minimal value,  $D_{\min}$ , from all data and order the data set ( $n = 1$  to  $N$ ); that is,  $D[1] = D_{\max}$  and  $D[N] = D_{\min}$ . Finally, we get  $D[N/2]$  as the medial value. Sets of  $D[N/2]$  are plotted in Figures 4 and 5.

## References

- [1] J. Velagic, B. Lacevic, and B. Perunicic, "A 3-level autonomous mobile robot navigation system designed by using reasoning/search approaches," *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 989–1004, 2006.
- [2] T. Miyake, H. Ishihara, R. Shoji, and S. Yoshida, "Development of small-size window cleaning robot a traveling direction control on vertical surface using accelerometer," in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '06)*, pp. 1302–1307, Luoyang, China, June 2006.
- [3] H. Kawata, A. Ohya, S. Yuta, W. Santosh, and T. Mori, "Development of ultra-small lightweight optical range sensor system," in *Proceedings of the IEEE IRS/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 1078–1083, August 2005.
- [4] P. Gemeiner and M. Vincze, "Motion and structure estimation from vision and inertial sensor data with high speed CMOS camera," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1853–1858, Barcelona, Spain, April 2005.
- [5] H. Zhang, S. Liu, and S. X. Yang, "A hybrid robot navigation approach based on partial planning and emotion-based behavior coordination," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 1183–1188, Beijing, China, October 2006.
- [6] J. Snyder, Y. Silverman, Y. Bai, and M. A. Maclver, "Underwater object tracking using electrical impedance tomography," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '12)*, pp. 520–525, Vilamoura, Portugal, October 2012.
- [7] D. F. Wolf, G. S. Sukhatme, D. Fox, and W. Burgard, "Autonomous terrain mapping and classification using hidden Markov models," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2026–2031, Barcelona, Spain, April 2005.
- [8] H. J. Kwak, D. H. Lee, J. M. Hwang, J. H. Kim, C. K. Kim, and G. T. Park, "Improvement of the inertial sensor-based localization

- for mobile robots using multiple estimation windows filter,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '12)*, pp. 876–881, Vilamoura, Portugal, October 2012.
- [9] D. H. Kim and S. Shin, “Local path planning using a new artificial potential function composition and its analytical design guidelines,” *Advanced Robotics*, vol. 20, no. 1, pp. 115–135, 2006.
- [10] M. Madry, C. H. Ek, R. Detry, K. Hang, and D. Kragic, “Improving generalization for 3D object categorization with global structure histograms,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '12)*, pp. 1379–1386, Vilamoura, Portugal, October 2012.
- [11] Y. Hirata, A. Hara, and K. Kosuge, “Motion control of passive intelligent walker using servo brakes,” *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 981–990, 2007.
- [12] Y. Hirata, A. Hara, and K. Kosuge, “Passive-type intelligent walking support system ‘RT Walker,’” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, pp. 3871–3876, October 2004.
- [13] O. Y. Chuy Jr., Y. Hirata, Z. Wang, and K. Kosuge, “A control approach based on passive behavior to enhance user interaction,” *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 899–908, 2007.
- [14] A. Kim and R. M. Eustice, “Real-time visual SLAM for autonomous underwater hull inspection using visual saliency,” *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 719–733, 2013.
- [15] <http://pdf1.alldatasheet.com/datasheet-pdf/view/166476/AD/ADUC7027.html>.
- [16] <http://www.findthatpdf.com/search-4444820-hPDF/download-documents-sharpgp2d12-15.pdf.htm>.
- [17] S. L. Smith, M. Schwager, and D. Rus, “Persistent robotic tasks: monitoring and sweeping in changing environments,” *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 410–426, 2012.
- [18] H. G. Tanner and A. Boddu, “Multiagent navigation functions revisited,” *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1346–1359, 2012.
- [19] N. Hunger, M. Baumann, J. A. Long, and J. Troccaz, “A 3-D ultrasound robotic prostate brachytherapy system with prostate motion tracking,” *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1382–1397, 2012.
- [20] A. Dietrich, T. Wimboeck, A. Albu-Schaeffer, and G. Hirzinger, “Integration of reactive, torque-based self-collision avoidance into a task hierarchy,” *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1278–1293, 2012.
- [21] P. Reist and R. D’Andrea, “Design and analysis of a blind juggling robot,” *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1228–1243, 2012.
- [22] V. Chalvet, Y. Haddab, and P. Lutz, “A microfabricated planner design microrobot for precise positioning based on bistable modules,” *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 641–649, 2013.
- [23] N. Rojas and F. Thomas, “The univariate closure conditions of all fully parallel planar robots derived from a single polynomial,” *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 758–765, 2013.
- [24] N. Dantam and M. Stilman, “The motion grammar: analysis of a linguistic method for robot control,” *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 704–718, 2013.

## Research Article

# Decision Making in Reinforcement Learning Using a Modified Learning Space Based on the Importance of Sensors

**Yasutaka Kishima, Kentarou Kurashige, and Toshihisa Kimura**

*Muroran Institute of Technology, 27-1 Mizumoto, Hokkaido, Muroran 0508585, Japan*

Correspondence should be addressed to Kentarou Kurashige; [kentarou@csse.muroran-it.ac.jp](mailto:kentarou@csse.muroran-it.ac.jp)

Received 15 March 2013; Accepted 21 May 2013

Academic Editor: Guangming Song

Copyright © 2013 Yasutaka Kishima et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many studies have been conducted on the application of reinforcement learning (RL) to robots. A robot which is made for general purpose has redundant sensors or actuators because it is difficult to assume an environment that the robot will face and a task that the robot must execute. In this case, Q-space on RL contains redundancy so that the robot must take much time to learn a given task. In this study, we focus on the importance of sensors with regard to a robot's performance of a particular task. The sensors that are applicable to a task differ according to the task. By using the importance of the sensors, we try to adjust the state number of the sensors and to reduce the size of Q-space. In this paper, we define the measure of importance of a sensor for a task with the correlation between the value of each sensor and reward. A robot calculates the importance of the sensors and makes the size of Q-space smaller. We propose the method which reduces learning space and construct the learning system by putting it in RL. In this paper, we confirm the effectiveness of our proposed system with an experimental robot.

## 1. Introduction

In recent years, reinforcement learning (RL) [1] has been actively studied, and many studies on its application to robots have been conducted [2–4]. A matter of concern in RL is the learning time. In RL, information from sensors is projected onto a state space. A robot learns the correspondence between each state action in the state space and determines the best correspondence. When the state space expands according to the number of sensors, the number of correspondences learned by the robot is also increased. In addition, the robot needs considerable much experience in each state to perform a task. Therefore, learning the best correspondence becomes time-consuming.

To overcome this problem, many studies have investigated accelerated RL [5–15] for which there are two approaches: a multirobot system and autonomous construction of the state space. In the former approach, multiple robots exchange experience information [5–9], so that each robot augments its own knowledge. Therefore, in this system robots can find the best correspondence between each state

and action faster than an individual robot in a single-robot system. In addition Nishi et al. [10] proposed a learning method in which a robot learns behavior through observations of the behavior of other robots, constructing its own relationships between state and behavior. However, in this approach, a robot needs other robots with whom to exchange experience information, and hence, if there are no additional robots in the system, this approach becomes irrelevant. We focus on the state construction of a single robot.

In contrast to the above approaches, in the approach that applies autonomous state space construction [11–16], a single robot is sufficient. The robot constructs a suitable state space based on its experience. Moreover, it can reduce the state space and learn correspondences faster. However, in the studies on this approach, all the sensors installed in the robot were considered to be equally important, and their number of states was the same. The installed sensors, which influence how well a robot executes a task, can be divided into important and unimportant sensors according to the task to be performed. However, in this approach, the robot has

to learn using unnecessary inputs because all the sensors are considered equally important. For example, Takahashi et al. [16] proposed a state that is constructed autonomously from state space by incremental state segmentation. In this method, the division rule of state is applied to the consecutiveness of the sensor data. Ishiguro et al. [12] proposed a state construction method using empirically obtained perceivers (EOPs). These methods are not focused on the importance of each sensor for performing a task. In fact, although the sensors installed on a robot have varying levels of importance in terms of performing a task, few studies focused on this aspect. We focus on the importance of a sensor for a particular task and propose a novel efficient learning method.

In this paper, we propose a system in which the robot constructs a temporary Q-space for decision making based on which sensors are considered important for the execution of a particular task, which facilitates high-speed learning. Since very important sensors affect the performance of a task significantly, they should sense the environment circumstantially. Thus, the number of their states is increased. On the other hand, since less important sensors do not affect the performance of a task, they may sense their environment less exactly. Thus, the number of their states, which is determined based on the importance of the sensors, is decreased. In this study, the importance of the sensors is defined as the correlation between the sensor value and the reward.

A temporary Q-space is constructed from the Q-space of the robot based on the importance of the sensors. The number of states of the Q-space is the maximum number that the sensors can describe. A Q-space is reduced by merging Q-values according to the number of the states of the sensors. Using the reduced Q-space, the robot can efficiently select an action, which is based on more information because the Q-values of low importance are merged. Therefore, the robot can learn correspondences using fewer experiences.

This system is effective for a variety of tasks. When this method is implemented, the amount of information that the robot requires in order to learn correspondences is reduced. As a result, when our proposed system is applied, a robot can learn correspondences faster than when an ordinary RL is applied.

## 2. Concept of Importance of Sensors

To select the sensors that are important to a certain task, a robot needs to measure the importance of each sensor. We focus on the correlation of the sensor value and the reward, which is specific to each task, as the measure of the importance of each sensor for a task. For example, in a garbage collection task, a robot is expected to approach a garbage heap and lift it. For this task, the reward is expressed by the distance between the robot and garbage heap and increases as the robot moves closer to the garbage heap. This implies that there is a correlation between the reward and the distance between the robot and garbage heap.

We show in Figure 1 an outline of the determination of the importance of the sensors where two types of sensor are installed on the robot. Via its sensors, the robot recognizes its

environment, which is expressed as a group of all the sensor values. The robot collects the sensor value of each sensor and the reward of the task to be performed and then determines the correlation between them. In Figure 1, the robot conducts this determination for sensors 1 and 2. The robot estimates the importance of the sensors according to the two types of correlation between the sensor value and reward: negative and positive.

Very important sensors affect the performance of a task. Therefore they should sense their environment circumstantially, and thus the number of their states is increased. On the other hand, less important sensors do not affect the performance of a task, and therefore they may sense the environment less exactly, and thus, the number of their states is decreased. The number of states is therefore determined based on the importance of the sensors.

## 3. Decision Making in Reinforcement Learning Using a Q-Space Based on the Importance of Sensors

*3.1. Outline of the Proposed System.* We show our proposed system in Figure 2. In the figure, the proposed system is divided into two stages. The first stage constitutes the proposed method whereby the robot determines the importance of its sensors. The next stage is RL.

In the first stage, the robot calculates the importance of the sensors for a task based on the correlations between each sensor and reward. The robot first estimates each sensor value and the reward and then calculates the coefficient of the correlation between them. Finally, it determines the important sensors based on this coefficient of correlation.

In RL, a robot learns the actions that are suitable for each state. This stage consists of an action evaluation element, wherein a temporary Q-space is constructed based on the determination of important sensors, and an action selection element. In the action evaluation element, each pair of state and action is evaluated and updated. A state contains the value of all the sensors. The robot constructs a temporary Q-space, by adding to it only those sensors that have been determined to be important. In the action selection element, the robot selects the action for a state recognized by the sensor based on the temporary Q-space.

We show the workflow of our proposed system in Figure 3. This workflow is executed by the robot for each action. We define this flow as one trial.

*3.2. Determination of the Importance of Sensors.* In the proposed method, the robot determines the importance of its sensors based on the correlation between each sensor value and reward, which it experiences. The robot stores each sensor value and the averaged reward as a list called a knowledge list, an example of which is shown in Figure 4. When a robot identifies a state that is not in the list, it adds it to the list. Then, the robot calculates the averaged reward and adds it to the list. On the other hand, when a recognized state is already in the list, it calculates the averaged reward and updates the list.

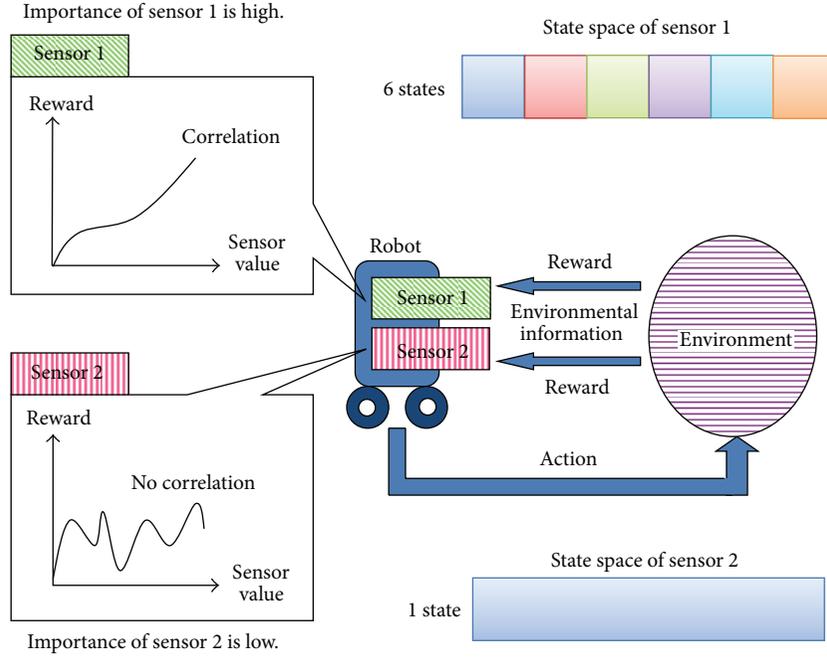


FIGURE 1: Outline of determination of importance.

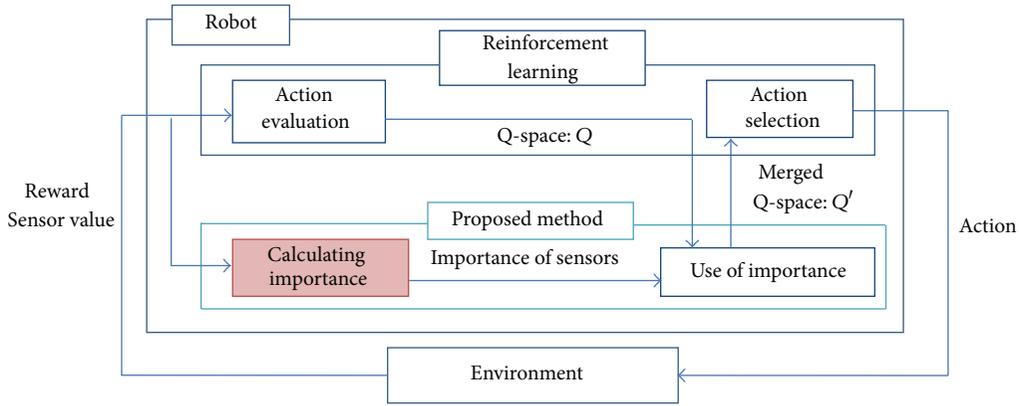


FIGURE 2: Outline of the proposed system.

A state  $s_i$  is defined as (1), where  $i$  is the state ID in Figure 4,  $e_{i,j}$  is the value of sensor  $j$ , and  $E_j$  is a group of values that describe a sensor:

$$s_i = \{e_{i,1}, e_{i,2}, \dots, e_{i,j}, e_{i,n} \mid e_{i,1} \in E_1, e_{i,2} \in E_2, \dots, e_{i,j} \in E_j, e_{i,n} \in E_n\}. \quad (1)$$

In this study, the rewards for the state experienced by the robot are weight averaged. Weighted averaging gives a greater weight to more recently obtained rewards. The averaged reward in state  $s_i$  is denoted by  $r'_t(s_i)$  and it is updated as (2), where  $r_t(s_i)$  is the reward obtained by the robot at time  $t$ :

$$r'_t(s_i) \leftarrow r_t(s_i) + \alpha_{ave} (r_t(s_i) - r'_{t-1}(s_i)). \quad (2)$$

The robot calculates the importance of each sensor based on the knowledge list. We use an equation for multiple coefficient correlation to calculate the sensor's level of importance.

In multiple coefficients, each partial regression coefficient represents the importance of each sensor. A sensor with a higher regression coefficient has a higher importance level.

The multiple regression equation is defined by (3), where  $a_{t,1}, a_{t,2}, \dots, a_{t,j}$  are the regression coefficients for each sensor,  $r'_t(t, s)$  is the averaged reward in state  $s$ , and  $a_0$  is the constant term:

$$r'_t(s_i) = a_1 e_{i,1} + a_2 e_{i,2} + \dots + a_i e_{i,n} + a_0. \quad (3)$$

A robot needs to calculate the regression coefficients of each sensor to calculate its importance. Each regression coefficient  $a_1, a_2, \dots, a_n$  is the solution resulting from the multiple simultaneous equation (4). Here,  $b_{ij}$ , calculated by (5), is the covariance of sensor  $i$  and sensor  $j$ , and  $b_i^2$ , calculated by (6), is the variance of sensor  $i$ . The average of sensor  $i$  values is calculated by (7), and the  $\bar{e}_i$  and  $\bar{e}_j$  in (5) are

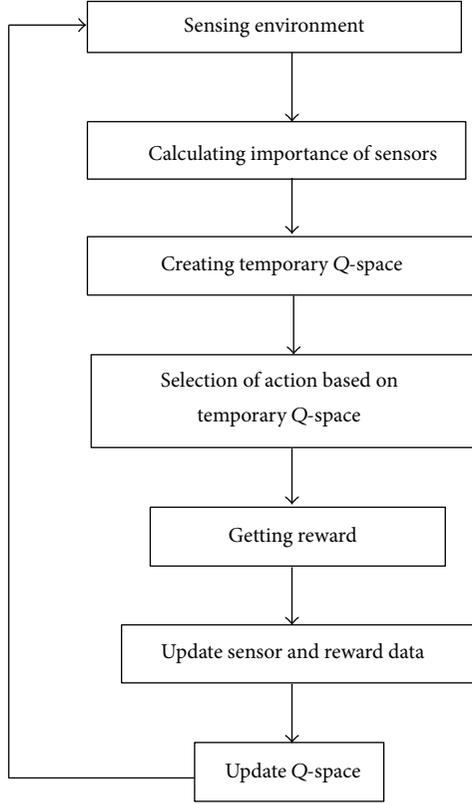


FIGURE 3: Flow of the proposed system.

State ID	Sensor value of sensor 1	Sensor value of sensor 2	...	Sensor value of sensor $n$	Averaged reward $\bar{r}'(s_i)$
1	10	56	...	111	45
2	23	45	...	123	23
...	...	...	...	...	...
$m$	90	10	...	15	132

FIGURE 4: List for calculating correlation.

calculated by (7).  $b_{ir}$ , calculated by (8), is the covariance of sensor  $i$  and reward  $r$ , and  $\bar{r}'$ , calculated by (9), is the average value of the averaged reward:

$$\begin{aligned}
 a_1 b_{11}^2 + a_2 b_{12} + \dots + a_n b_{1n} &= b_{1r}, \\
 a_1 b_{12} + a_2 b_{22}^2 + \dots + a_n b_{2n} &= b_{2r}, \\
 &\vdots
 \end{aligned} \tag{4}$$

$$a_1 b_{1n} + a_2 b_{2n} + \dots + a_n b_{nn}^2 = b_{nr},$$

$$b_{ij} = \frac{1}{m} \sum_{k=1}^m (e_{i,k} - \bar{e}_i)(e_{j,k} - \bar{e}_j), \tag{5}$$

$$b_i^2 = \frac{1}{m} \sum_{k=1}^m (e_{i,k} - \bar{e}_i)^2, \tag{6}$$

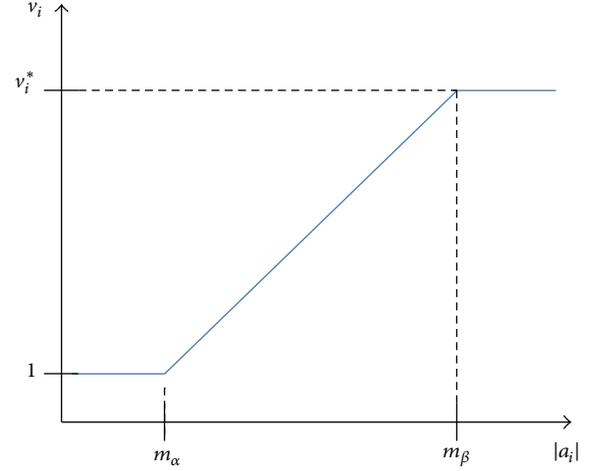


FIGURE 5: Variation property.

$$\bar{e}_j = \frac{1}{m} \sum_{k=1}^m e_{j,k}, \tag{7}$$

$$b_{ir} = \frac{1}{m} \sum_{k=1}^m (e_{i,k} - \bar{e}_i)(r_k - \bar{r}'), \tag{8}$$

$$\bar{r}' = \frac{1}{m} \sum_{k=1}^m r'(s_k). \tag{9}$$

### 3.3. Determination of the Number of States of Each Sensor.

The number of states is determined based on the number of states of each sensor, which is determined based on the regression coefficient of each sensor. When the absolute value of the regression coefficient of a sensor is higher, the number of states of the sensor is increased.

We use a variation property, shown in Figure 5, as the number of states based on the regression coefficient. When the regression coefficient of a sensor is less than  $m_\beta$ , the minimum number of states is 1; when it is greater than  $m_\alpha$ , the maximum number of states is  $v_i^*$ ; and when it is from  $m_\alpha$  to  $m_\beta$ , the number of state increases gradually. The parameters  $m_\beta$  and  $m_\alpha$  are determined by a human. The formulation of the property is determined by (10).  $v_i$  is the number of states of sensor  $i$ :

$$v_i = \begin{cases} 1 & (|a_i| < m_\alpha), \\ \left\lceil \frac{v_i^* - 1}{m_\beta - m_\alpha} |a_i| + \frac{m_\beta - m_\alpha v_i^*}{m_\beta - m_\alpha} \right\rceil & (m_\alpha \leq |a_i| \leq m_\beta), \\ v_i^* & (m_\beta < |a_i|). \end{cases} \tag{10}$$

Here,  $v_i^*$  is determined based on the performances of the sensor. We focus on the resolution and maximum range as the performance of the sensor. When the performance of the resolution and maximum range of a sensor are high, the robot can describe more states.  $v_i^*$  is the state number, which can be calculated based on the resolution and maximum range of the sensor, as in (11), where  $g_{\max,i}$  is the maximum range of sensor

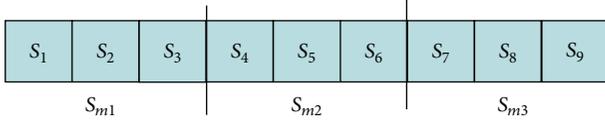


FIGURE 6: Example of merged state of a sensor.

$i$ ,  $g_{\min,i}$  is the minimum range of sensor  $i$ ,  $b_i$  is resolution of sensor  $i$ , and  $v_i^*$  is calculated for each sensor. We define a state of a sensor on  $v_i^*$  as a “state unit”:

$$v_i^* = \frac{g_{\max,i} - g_{\min,i}}{b_i}. \quad (11)$$

**3.4. Construction of a Temporary Q-Space for Action Selection.** The robot constructs a temporary Q-space based on  $v_i$ . The temporary Q-space consists of a unit, which is several state units merged together. We show an example of a merged state space in Figure 6. In this example,  $v_i^*$  is 9 and the number of state units is 9. When  $v_i$  is 3, the temporary Q-space is constructed of three states obtained by merging three state units. This example is focused on one sensor. All the installed sensors are merged according to  $v_i$ .

Here, when the number of states is  $v_i$ , the number of state units in a merged state  $v_i'$  is calculated by the following equation:

$$v_i' = \left\lceil \frac{v_i^*}{v_i} \right\rceil. \quad (12)$$

The Q-value of each state unit in merged targets is averaged, as shown in Figure 7, where an example of the temporary Q-space when  $v_1^*$  and  $v_2^*$  are 6,  $v_1'$  is 3, and  $v_2'$  is 3 is depicted. For sensor  $S_1$ , each of the states is three merged state units. For sensor  $S_2$ , each of the states is two merged state units. Merged state units are averaged Q-values.

The robot selects an action based on the temporary Q-space for the current state. It recognizes the current state as state unit  $s$ . When the current state is  $s$ , the merged target group of the state unit is  $C_i = \{v_{i,o}, v_{i,p}, \dots, v_{i,q}\}$ , where  $v_{i,o}, v_{i,p}, \dots, v_{i,q}$  in the group are the state units of a sensor. Similarly, the target group of sensor  $t$  is  $C_t = \{v_{t,o}, v_{t,p}, \dots, v_{t,q}\}$ , and the target group of sensor  $u$  is  $C_u = \{v_{u,o}, v_{u,p}, \dots, v_{u,q}\}$ . Here, the Q-value of the merged target in Q-space  $Q(s_m, i)$  is defined by (13), where  $R(s_m, a)$  is the total reward of merged state units, defined as (14), where  $R(s_w, a)$  is the total reward at state  $s_w = \{v_{i,w}, v_{t,w}, \dots, v_{u,w}\}$  and action  $a$  and  $E(s_m, a)$  is the total number of the experiences of the state action pair  $(s_m, a)$ , defined as (15). Merging is performed for all the actions in state  $s_m$ :

$$Q(s_m, a) = \frac{R(s_m, a)}{E(s_m, a)}, \quad (13)$$

$$R(s_m, a) = \sum_{v_{i,w} \in C_i} \sum_{v_{t,w} \in C_t} \dots \sum_{v_{u,w} \in C_u} (Q(s_w, a) \cdot E(s_w, a)), \quad (14)$$

$$E(s_m, a) = \sum_{v_{i,w} \in C_i} \sum_{v_{t,w} \in C_t} \dots \sum_{v_{u,w} \in C_u} E(s_w, a). \quad (15)$$

**3.5. Action Selection.** The robot selects an action based on the temporary Q-space. We apply the  $\epsilon$ -greedy method for action selection. This method selects the action that has the highest Q-value in the current state unit  $s$ . However, the method selects an action randomly with probability  $\epsilon$ .

**3.6. Action Evaluation.** We apply the weighted averaging method as the action evaluation method. This method evaluates actions by assigning a weight to each reward recently obtained by the robot. When the current state unit of the robot is  $s$  and the selected action is  $a$ , Q-value ( $Q(s, a)$ ) is updated by (16).  $\alpha$  is a step size parameter ( $0 \leq \alpha \leq 1$ ):

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r - Q(s, a)]. \quad (16)$$

## 4. Experiment to Confirm the Effectiveness of the Proposed System

**4.1. Outline of the Experiment.** In this section, we describe our evaluation of the effectiveness of the proposed system via an experimental robot. The experimental environment is shown in Figure 8. This environment is surrounded by walls of length 1100 mm. We prepared an experimental robot, as shown in Figure 9. The robot has two distance sensors, which measure the distance between the current position of the robot and the walls. It can recognize the current state of a sensor as the state value  $u$ , as shown in Figure 10. Its sensors are divided into 11 states every 70 mm, and each state is given a state value. In this experiment, each sensor had 11 states and the total number of states was 121. The robot has omnivheels and can move in the forward, back, left, right, and each diagonal direction but cannot turn. It can move 70 mm in one action when it is not moving in a diagonal direction. When the robot moves in a diagonal direction, it moves in the cross direction and then in the lengthwise direction.

In the experiment, the task of the robot was to move close to wall A. The robot could obtain rewards according to its distance only from wall A. It was placed at the lower right corner of the environment and when it reached wall A, an episode was considered. The experiment was concluded at  $n_e$  episodes.

This task seems simple at first glance. However, it is difficult for an RL robot. In this task, when a robot strays into an unexperienced area, its action selection becomes random, because the Q-values of the states in the area are at their initial value. Therefore, the robot strays more by repeating a random action selection and takes time to estimate the Q-values of the states and to leave the area. In addition, in this experiment, the task was being performed by an experimental robot. Therefore, it is possible that the sensor noise affected its learning of, for example, gap state recognition.

In real world use scenes, a robot is rarely required to move in a maze-like environment with many obstacles. Most use scenes are open spaces, such as a warehouse or park. Therefore, the environment we adopted is appropriate for this experiment.

We confirmed the effectiveness of the proposed system by comparing it with a conventional RL. Therefore, we prepared two types of robot for comparison, to one of which the

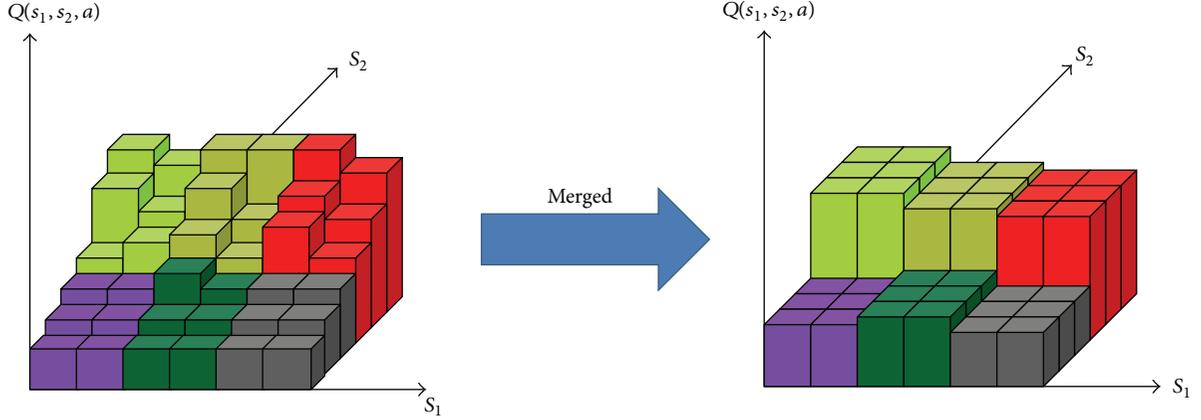


FIGURE 7: Example of projection for a temporary Q-space.

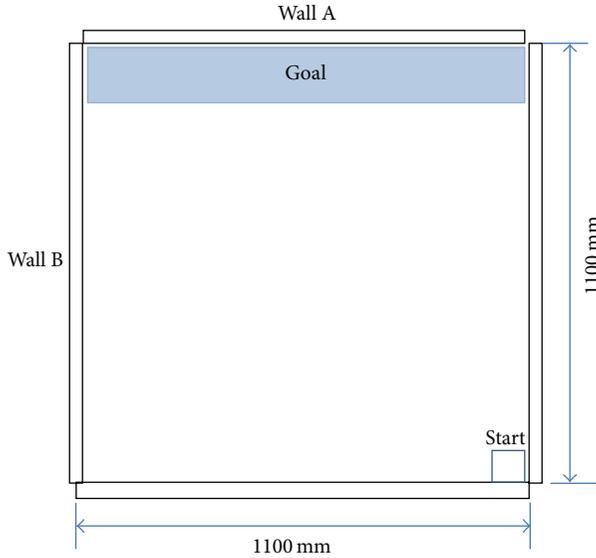


FIGURE 8: Experimental environment.

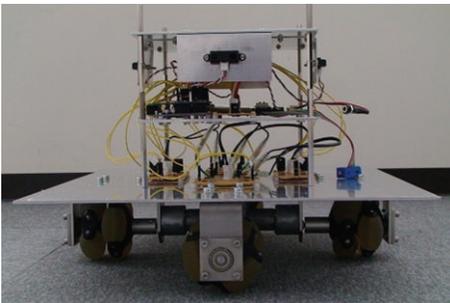


FIGURE 9: Experimental robot.

proposed system was applied, and to the other RL. The number of states of each sensor of the robot to which RL was applied was  $v^*$ . We compared these agents in terms of the total reward obtained after  $n_e$  episodes were completed.

TABLE 1: Experimental settings.

$n_e$	100
$\alpha_{ave}$	0.1
$\epsilon$	0.1
$\alpha$	0.1
Initial value of Q-value	0
Maximum range of sensors $g_{max}$	1100 mm
Minimum range of sensors $g_{min}$	0
Resolution of sensors $b$	100 mm
$m_\alpha$	0.2
$m_\beta$	0.8
The number of initial states	11
Initial importance	1.0

**4.2. Experimental Setup.** In this section, we explain the reward for each task and discuss the parameter settings. In task A, the robot can obtain a higher reward by increasing the distance between its current position and wall A, as defined in (17), where  $d_A$  is the state value shown in Figure 10.  $d_A$  is determined based on the actual measurement value of the distance between the current position and wall A:

$$r = 11 - d_A. \quad (17)$$

A list of the parameter settings of this experiment is given in Table 1. In this experiment, the settings of the maximum range, minimum range, and resolution of sensors remained the same.  $v^*$  was 11 according to (11). When the robot started a new episode, the Q-space and state knowledge list from previous episodes were adopted.

**4.3. Experimental Results.** The experimental results are shown in Figures 11–15. Figure 11 shows the importance of the sensors for the final action in each episode. The importance of the sensors is represented by the regression coefficients. In first episode, the regression coefficients of each sensor are converged in the early phase of learning. The regression of the coefficient of sensor A is greater than the threshold  $m_\beta = 0.8$ . On the other hand, the regression of the coefficient of sensor

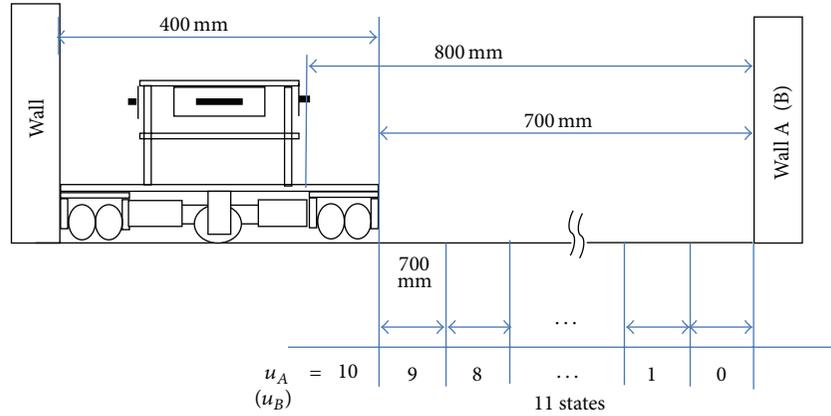


FIGURE 10: State recognition of the robot.

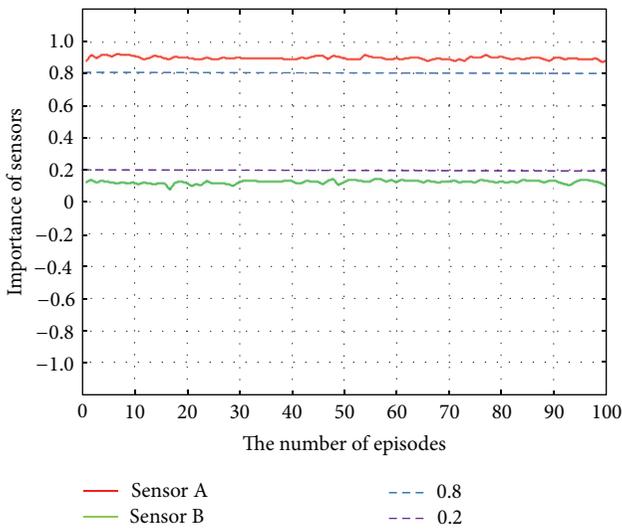


FIGURE 11: Importance of sensors in each episode.

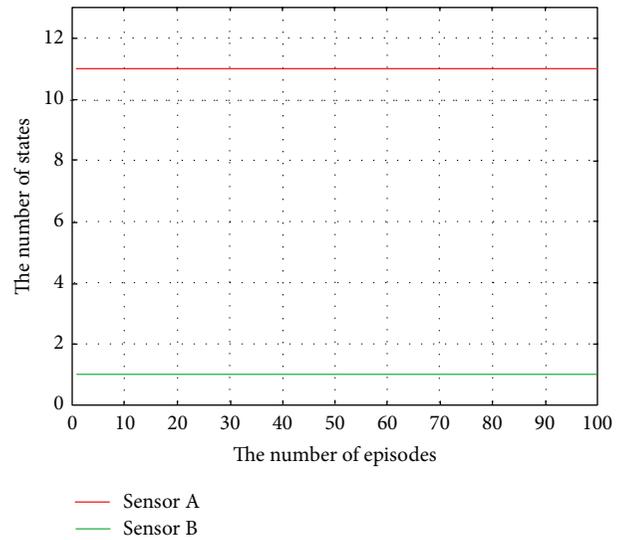


FIGURE 12: The number of states in each episode.

B is smaller than the threshold  $m_\alpha = 0.2$ . In this task, only wall A is related to reward and its importance is high. It is valid that the regression of the coefficient of the wall A sensor is high and that of wall B is low.

Figure 12 shows the number of the states of the sensors in the final action in each episode. The number of states of sensor A is the maximum, 11. The number of states of sensor B is the minimum, 1. Using these results, the robot can construct a correct temporary Q-space.

Figures 13 and 14 show the importance of the sensors and the number of states in each action in the first episode, respectively. Until the 30th action, the importance of the sensors is unstable. The reason is that the robot has insufficient knowledge to calculate the regression of the coefficient correctly. After the 31st action, the robot has sufficient knowledge and can therefore calculate the regression of the coefficient correctly. Until the 30th action, because of the instability in the number of the states of the sensors, the

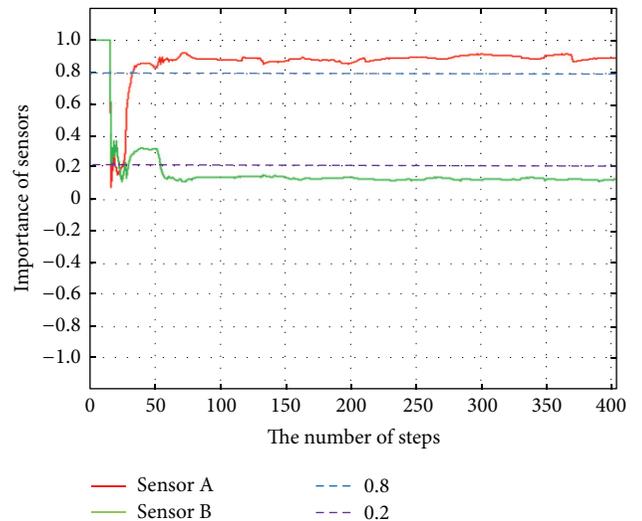


FIGURE 13: Importance of sensors in each action in episode 1.

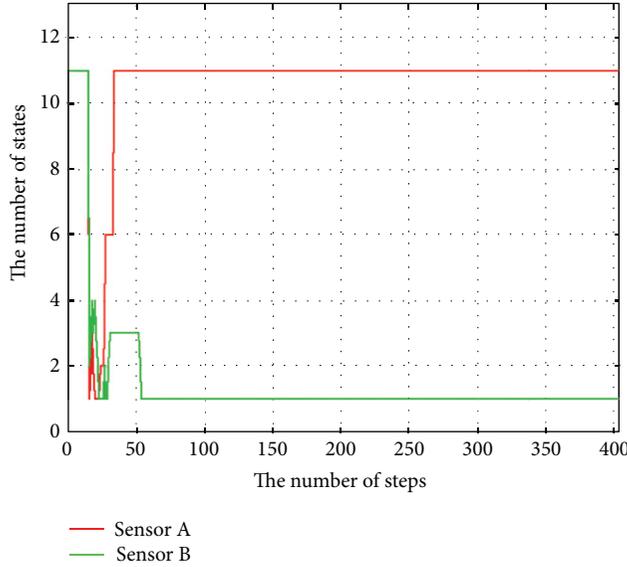


FIGURE 14: The number of states in each action in episode 1.

robot's ability to calculate the importance of the sensors is negatively affected by its insufficient knowledge.

Figure 15 shows the total number of actions in each episode. The robot using the proposed method achieves a high convergence of the number of actions as compared to that using RL. This is because the robot using RL strays into an area for which it has no experience and takes time to leave it. On the other hand, this does not occur when the proposed method is used. This is because, when the robot strays into an area for which it has no experience, it can use other  $Q$ -values learned in other states based on the constructed temporary  $Q$ -space, which consists of only important sensors. Therefore, the robot does not need to take time for learning in the unexperienced area. Thus, the robot focuses on only the important sensors and selects the suitable action by using this  $Q$ -space. These results show that the proposed method is effective for learning.

## 5. Conclusion

In this paper, we proposed a method in which a robot selects an action by using a temporary  $Q$ -space based on the importance of its sensors. This method assumes that there is a correlation between the sensor value and reward. The robot calculates the regression coefficient using a multiple regression equation of the sensor value and reward. The robot determines the importance of its sensors according to the regression coefficient. The higher the level of importance, the larger the number of states. To select an action, the robot constructs a temporary  $Q$ -space based on the importance of the sensors. It then selects actions based on the temporary  $Q$ -space. Thus, the robot is able to learn faster.

We examined the effectiveness of the proposed system using an experimental robot. We investigated a task for the execution of which only one sensor of a sensor pair was important. We compared the proposed system with

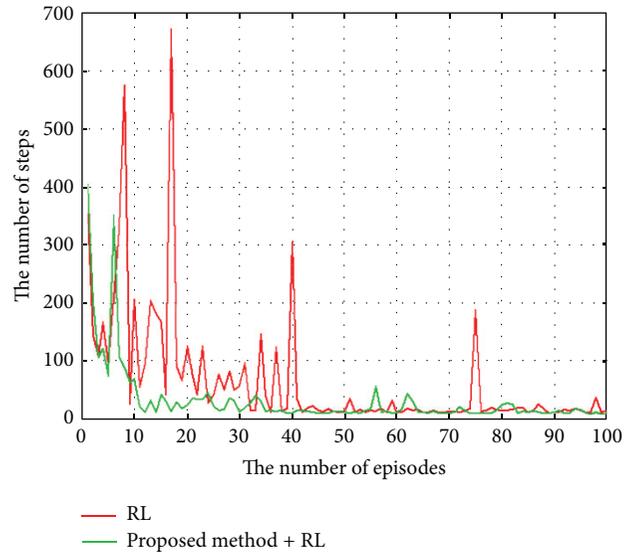


FIGURE 15: The number of total actions at each episode.

conventional RL. The robot used sensors whose number of states is common in the case of conventional RL.

The results showed that using the proposed system the robot could calculate the importance of the sensors correctly. In addition, convergence speed was faster than that in conventional RL. Thus, we confirmed the effectiveness of the constructed system and the proposed method.

In future studies, first, we will examine the effectiveness of the proposed system by comparing it with other autonomous state construction systems. In this study, we examined only normal reinforcement learning. It is necessary to examine the proposed system by comparing it with those proposed in related studies. Then, we will modify the proposed method. Currently, the proposed method cannot be applied in a delayed reward task, because the regression coefficient is calculated using the immediate reward in each state. When a task is a delayed reward task, the robot cannot calculate the importance of its sensors. Therefore, we will modify the proposed method so that it can be applied in a delayed reward task by using information that does not include the reward.

## References

- [1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.
- [2] T. Kondo and K. Ito, "A reinforcement learning with evolutionary state recruitment strategy for autonomous mobile robots control," *Robotics and Autonomous Systems*, vol. 46, no. 2, pp. 111–124, 2004.
- [3] K. J. Person, E. Oztop, and J. Peters, "Reinforcement learning to adjust robot movements to new situations," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 2650–2655, 2010.
- [4] N. Navarro, C. Weber, and S. Wermter, "Real-world reinforcement learning for autonomous humanoid robot charging in a home environment," in *Towards Autonomous Robotic Systems*,

- vol. 6856 of *Lecture Notes in Computer Science*, pp. 231–240, Springer, 2011.
- [5] M. Tan, “Multi-agent reinforcement learning: independent vs. cooperative agents,” in *Proceedings of the 10th International Conference on Machine Learning*, 1993.
- [6] M. N. Ahmadabadi, M. Asadpur, S. H. Khodaabakhsh, and E. Nakano, “Expertness measuring in cooperative learning,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)*, vol. 3, pp. 2261–2267, November 2000.
- [7] M. N. Ahmadabadi and M. Asadpour, “Expertness based cooperative Q-learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 32, no. 1, pp. 66–76, 2002.
- [8] H. Iima and Y. Kuroe, “Swarm reinforcement learning algorithm based on exchanging information among agents,” *Transactions of the Society of Instrument and Control Engineers*, vol. 42, no. 11, pp. 1244–1251, 2006.
- [9] Y. Yongming, T. Yantao, and M. Hao, “Cooperative Q learning based on blackboard architecture,” in *Proceedings of International Conference on Computational Intelligence and Security Workshops (CIS '07)*, pp. 224–227, December 2007.
- [10] T. Nishi, Y. Takahashi, and M. Asada, “Incremental behavior acquisition based on reliability of observed behavior recognition,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 70–75, November 2007.
- [11] M. Asada, S. Noda, and K. Hosoda, “Action-based sensor space categorization for robot learning,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '96)*, pp. 1502–1509, November 1996.
- [12] H. Ishiguro, R. Sato, and T. Ishida, “Robot oriented state space construction,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '96)*, pp. 1496–1501, November 1996.
- [13] K. Samejima and T. Omori, “Adaptive internal state space construction method for reinforcement learning of a real-world agent,” *Neural Networks*, vol. 12, no. 7-8, pp. 1143–1155, 1999.
- [14] A. J. Smith, “Applications of the self-organising map to reinforcement learning,” *Neural Networks*, vol. 15, no. 8-9, pp. 1107–1124, 2002.
- [15] K. T. Aung and T. Fuchda, “A proposition of adaptive state space partition in reinforcement learning with voronoi tessellation,” in *Proceedings of the 17th International Symposium on Artificial Life and Robotics*, pp. 638–641, 2012.
- [16] Y. Takahashi, M. Asada, and K. Hosoda, “Reasonable performance in less learning time by real robot based on incremental state space segmentation,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 96)*, pp. 1518–1524, November 1996.

## Research Article

# Multiple Harmonics Fitting Algorithms Applied to Periodic Signals Based on Hilbert-Huang Transform

Hui Wang,<sup>1</sup> Zhengshi Liu,<sup>2</sup> Bin Zhu,<sup>3</sup> and Qianjun Song<sup>1</sup>

<sup>1</sup> Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China

<sup>2</sup> Institute of Sound and Vibration Research, Hefei University of Technology, Hefei 230009, China

<sup>3</sup> Electronic Engineering Institute, Hefei 230037, China

Correspondence should be addressed to Hui Wang; wanghui@iim.ac.cn

Received 21 March 2013; Accepted 24 April 2013

Academic Editor: Aiguo Song

Copyright © 2013 Hui Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new generation of multipurpose measurement equipment is transforming the role of computers in instrumentation. The new features involve mixed devices, such as kinds of sensors, analog-to-digital and digital-to-analog converters, and digital signal processing techniques, that are able to substitute typical discrete instruments like multimeters and analyzers. Signal-processing applications frequently use least-squares (LS) sine-fitting algorithms. Periodic signals may be interpreted as a sum of sine waves with multiple frequencies: the Fourier series. This paper describes a new sine fitting algorithm that is able to fit a multiharmonic acquired periodic signal. By means of a “sinusoidal wave” whose amplitude and phase are both transient, the “triangular wave” can be reconstructed on the basis of Hilbert-Huang transform (HHT). This method can be used to test effective number of bits (ENOBs) of analog-to-digital converter (ADC), avoiding the trouble of selecting initial value of the parameters and working out the nonlinear equations. The simulation results show that the algorithm is precise and efficient. In the case of enough sampling points, even under the circumstances of low-resolution signal with the harmonic distortion existing, the root mean square (RMS) error between the sampling data of original “triangular wave” and the corresponding points of fitting “sinusoidal wave” is marvelously small. That maybe means, under the circumstances of any periodic signal, that ENOBs of high-resolution ADC can be tested accurately.

## 1. Introduction

For robot sensors, ADC plays a key role, which receives the signal coming from front-end circuit and then converts it into digital logic output [1, 2]. ADC testing frequently uses periodic signals to obtain most of the specification parameters, such as the ENOBs, signal-to-noise and distortion (SINAD) ratio, integral nonlinearity (INL), differential nonlinearity (DNL), and the transfer function. Sine fitting is a very efficient and fast way to help in the evaluation of most of these characteristic parameters. IEEE standards 1057 [3] and 1241 [4] present two methods that estimate three (amplitude, phase, and dc component) or four parameters (including also the frequency) of a sine wave that best fit a set of acquired samples. The two sine fitting methods described by IEEE standards are general and classical methods aiming at the data acquisition channel as well as the ADC demarcation.

However, as for a sinusoidal sampling signal with harmonic distortion, whether using the classical three-parameter or four-parameter sine fitting methods, both of them will result in biased estimation [5].

But it is impractical to make use of one sinusoid for the evaluation of ADC when input signal is a harmonic signal or a triangular wave. The difference between a simple sine wave and a harmonic signal or the difference between a simple sine wave and a triangular wave is obviously great. Errors can arise in frequency, amplitude, and phase [6].

A maximum likelihood improved sine-wave fitting procedure for characterizing data acquisition (DAQ) channels and ADC, which is also capable of fitting multiple harmonics of an input signal, was presented in [5]. A different method was proposed in [7] based on the spectral analysis and fitting by interpolating the acquired samples. Moreover, several other

algorithms were proposed to fit multiple harmonics in [8–10]. However, the application of those methods is much complex.

Next, some relatively simpler method presented in document [6, 11] using the IEEE standards was proposed to best fit multiple harmonics particularly. This is an LS sine-fitting algorithm based on four-parameters sine fitting at first and three-parameters sine fitting afterwards aiming at the periodic signal containing multiple harmonics.

It has been shown by the experimental results that the LS sine-fitting algorithm can fit up to 43 harmonics of a triangular wave with very good results. The dimension of computer memory limits the highest harmonic that can be considered in this method. It should be noted that there is no algorithm-related limitation regarding the number of harmonics or the highest possible harmonic to consider. The limitations are concerned with memory requirements and convergence of the nonlinear least-square method, as in [3, 4].

In the LS sine-fitting algorithm a problem is the evaluation of the initial condition of the fundamental harmonic frequency, to guarantee the convergence. To overcome this problem, a method devoted to the improved evaluation of initial condition for the LS sine-fitting algorithm is presented in [12]. In particular, the method is based on the algebraic derivative approach in the frequency domain.

The Hilbert-Huang transform (HHT), which was first developed by Huang et al. at the end of the last century, is a novel signal analysis method. The HHT is derived from the principals of empirical mode decomposition (EMD) and Hilbert transform (HT) [13, 14]. With EMD, any complicated data set can be decomposed into finite (often less) number of intrinsic mode functions (IMFs) which admit well-behaved Hilbert transform (HT). With HT, each IMF yields its instantaneous frequency (IF), phase, and envelope as functions of time. The wavelet methods are also popular approaches for performing both decomposition and reconstruction for signals and analyzing the frequency of signals, but the results of wavelet transform depend on the selection of wavelet base [15]. In contrast, the HHT method is self-adaptive, unnecessary to select the base functions, and the results are more stable.

In this paper, one “sinusoidal” fitting method to the original triangular wave which is based on the HHT will be put forward, which can avoid the trouble of selecting the initial value of the parameters and working out the nonlinear equations compared to the LS sine-fitting algorithm. First of all, the EMD process is carried out on a triangular wave, and the originally got first-order IMF is used to reconstruct the triangular wave. Secondly, the HT process is carried out on the first-order IMF, then amplitude function and phase function of analytic signal corresponding to the reconstructed “triangular wave” are obtained, and so, the key parameters of the “sinusoidal curve” are achieved. The “sinusoidal curve” can simulate the original triangular wave very well, and its particularity lies in that the amplitude and phase of the “sinusoidal” curve are instantaneous; in other words, both the amplitude and phase vary with time. The simulation results show that, in the case of sufficient sampling points, the magnitude of the RMS error between the sampling

data of triangular wave and corresponding points of fitting “sinusoidal” curve model is quite small, which proves that good fit is realized.

The algorithm of multiple harmonics least-squares fitting to estimate the offset, fundamental frequency, and the amplitude and phase values of the harmonics of a multiharmonic signal is detailed in the second section. The algorithm of HHT fitting to get the instantaneous amplitude and phase of the “sinusoidal curve” is detailed in the third section. The fourth section describes the numerical simulations and results based on two algorithms, respectively.

## 2. Least-Squares (LS) Multiharmonic Fitting Method [6]

The sampled periodic signal may be described as a sum of sine waves with different frequencies, multiple of the fundamental ( $f$ ), individual amplitudes ( $D_h$ ), and phases ( $\phi_h$ ):

$$y(t) = C + \sum_{h=1}^H D_h \cos(2h\pi ft + \phi_h). \quad (1)$$

A set of  $M$  samples,  $(y_1, \dots, y_m, \dots, y_M)$ , is acquired at a sampling frequency  $f_s$ . To each sample, a relative timestamp is attributed according to the sample number and the sampling frequency:

$$t_m = \frac{m-1}{f_s}. \quad (2)$$

Independently of the number of harmonics  $H$  in the input signal, the method can determine only the amplitudes and phases of the first  $N$  ( $N \leq H$ ) harmonics.

And so, the residuals  $r_m$  of fitting are

$$r_m = y_m - \widehat{C} - \sum_{n=1}^N [\widehat{A}_n \cos(2n\pi \widehat{f} t_m) + \widehat{B}_n \sin(2n\pi \widehat{f} t_m)], \quad (3)$$

where  $\widehat{C}$  is the estimated dc component and  $\widehat{f}$  is the estimated frequency.  $\widehat{A}_n$  and  $\widehat{B}_n$  are the orthogonal estimated amplitudes of harmonic  $n$ . The amplitudes and the phases of the harmonics are determined by

$$\begin{aligned} \widehat{D}_n &= \sqrt{\widehat{A}_n^2 + \widehat{B}_n^2}, \\ \phi_n &= \arctan 2 \left( \widehat{B}_n, \widehat{A}_n \right). \end{aligned} \quad (4)$$

The LS method minimizes the sum of the residuals, while the total rms error is

$$\rho = \sqrt{\frac{1}{M} \sum_{m=1}^M r_m^2}. \quad (5)$$

The first step of the algorithm is to estimate the initial frequency  $\widehat{f}_0$  by an interpolated fast Fourier transform [16].

With this frequency, a set of initial estimated parameters are determined like in the three-parameter sine-fitting algorithm of [3, 4]. This estimation is done in one noniterative step, producing the harmonic amplitudes and phases that minimize the total rms error for the frequency  $\hat{f}_0$ .

The  $y$  matrix contains the sample values:

$$y = [y_1 \ \cdots \ y_m \ \cdots \ y_M]^T. \quad (6)$$

To determine the initial harmonic parameters, one two-column matrix for each harmonic:

$$W_n = [\cos(n\Omega) \ \sin(n\Omega)] \quad (7)$$

with  $\Omega_m = 2\pi\hat{f}_0 t_m$  is used.

To determine the estimated values of  $\hat{C}$ ,  $\hat{A}_n$ , and  $\hat{B}_n$ , the full LS matrix is

$$D = [1 \ W_1 \ \cdots \ W_n \ \cdots \ W_N]. \quad (8)$$

The LS solution vector

$$\hat{x}_0 = [\hat{C} \ \hat{A}_1 \ \hat{B}_1 \ \cdots \ \hat{A}_n \ \hat{B}_n \ \cdots \ \hat{A}_N \ \hat{B}_N]^T \quad (9)$$

is determined by

$$\hat{x}_0 = (D^T D)^{-1} (D^T y). \quad (10)$$

The solution vector contains the harmonic parameters that minimize the rms error (5) for the interpolated discrete Fourier transform (IpDFT) frequency  $\hat{f}_0$ .

To also determine the best frequency, an iterative method is required that can, in iteration  $i$ , adjust all the parameters  $C^i$ ,  $A_n^i$ ,  $B_n^i$ , and  $\omega^i = 2\pi f^i$ , which minimize the total residual error. In order to include the frequency, vector  $Q$

$$Q_m = \sum_{n=1}^N -A_n^{i-1} n t_m \sin(n\Omega_m^{i-1}) + B_n^{i-1} n t_m \cos(n\Omega_m^{i-1}) \quad (11)$$

with  $\Omega_m^{i-1} = \omega^{i-1} t_m$  is used in the new full LS matrix:

$$D^i = [Q \ 1 \ W_1 \ \cdots \ W_n \ \cdots \ W_N]. \quad (12)$$

The LS solution for iteration  $i$  is obtained by

$$\hat{x}^i = [(D^i)^T D^i]^{-1} [(D^i)^T y], \quad (13)$$

where the solution vector is now

$$\hat{x}^i = [\Delta\omega^i \ C^i \ A_1^i \ B_1^i \ \cdots \ A_n^i \ B_n^i \ \cdots \ A_N^i \ B_N^i]^T. \quad (14)$$

After each iteration, the frequency estimate is updated using  $\omega^i = \omega^{i-1} + \Delta\omega^{i-1}$ .

This process is repeated based on the new values of  $A_n^i$ ,  $B_n^i$ , and  $\omega^i$ . The iterations stop when the frequency changes are suitably small.

### 3. “Sinusoidal” Fitting Method Based on HHT

**3.1. EMD Method [13, 14].** EMD method is developed from the simple assumption that any signal consists of different simple intrinsic modes of oscillations. In this way, each signal could be decomposed into a number of IMFs, each of which must satisfy the following two conditions: (a) in the whole data set, the number of extrema and the number of zero-crossings must either equal or differ at most by one; (b) at any point, the mean value of the envelope defined by local maxima and the envelope defined by the local minima is zero.

An IMF represents a simple oscillatory mode compared with the simple harmonic function. With that definition, any signal  $s(t)$  can be decomposed as follows.

- (1) Identify all the local extrema of signal  $s(t)$ , and then connect all the local maxima by a cubic spline line as the upper envelope.
- (2) Repeat the procedure for the local minima to produce the lower envelope. The upper and lower envelopes should cover all the data between them.
- (3) The mean of upper and low envelope value is designated as  $m(t)$ ; and the difference between the signal  $s(t)$  and  $m(t)$  is  $h(t)$ , that is,  $s(t) - m(t) = h(t)$ .
- (4) If the sifting result  $h(t)$  is an IMF, stop. Otherwise, treat  $h(t)$  as the original signal and iterate on  $h(t)$  through Steps (1), (2), (3).
- (5) Denote by  $c_1(t)$  the first IMF and set  $r_1(t) = s(t) - c_1(t)$  the first residue. The algorithm proceeds to select the next IMF by applying the above procedure to the first residue  $r_1(t)$ . This process is repeated until the last residue  $r_n(t)$  has at most one extremum or becomes constant. The signal  $s(t)$  then can be represented as

$$s(t) = \sum_{i=1}^n c_i(t) + r_n(t). \quad (15)$$

As in Huang et al., the stopping condition in Step (4) is to limit the following standard deviation from two consecutive results in the sifting process:

$$SD \geq \sum_{i=0}^T \frac{[h_{m-1}(t) - h_m(t)]^2}{h_{m-1}(t)^2}, \quad (16)$$

where  $T$  is the length of the signal and  $h_m(t)$  is the sifting result in the  $m$ th iteration. A typical stopping value of SD is set between 0.2 and 0.3.

**3.2. HT.** After obtaining the IMFs, we apply the HT on each IMF:

$$H[c_i(t)] = c_i(t) * \frac{1}{\pi t} = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{c_i(\tau)}{t - \tau} d\tau, \quad (17)$$

where  $P$  denotes the principal Cauchy value.

By definition, the analytic signal corresponding to each IMF is

$$z_i(t) = c_i(t) + jH[c_i(t)] = a_i(t) e^{i\theta_i(t)}, \quad (18)$$

where  $a_i(t)$  and  $\theta_i(t)$  are the instantaneous amplitude and the instantaneous phase, respectively, of the signal at time  $t$ .

According to the analytic signal,  $c_i(t)$  can be represented as

$$c_i(t) = a_i(t) \cos \theta_i(t). \quad (19)$$

The instantaneous amplitude and instantaneous phase of the analytic signal are defined in the usual manner and can, respectively, be represented as

$$\begin{aligned} a_i(t) &= |z_i(t)| = \sqrt{c_i(t)^2 + H[c_i(t)]^2}, \\ \theta_i(t) &= \arg[z_i(t)] = \arctan\left(\frac{H[c_i(t)]}{c_i(t)}\right). \end{aligned} \quad (20)$$

The analytic signal represents the time-series as a slowly varying amplitude envelope modulating a faster varying phase function [17]. The IF is then given by

$$f_i(t) = \left(\frac{1}{2\pi}\right) \frac{d[\theta_i(t)]}{dt}. \quad (21)$$

3.3. “Sinusoidal Fitting” Process Based on HHT. The whole process of this method can be described as follows.

*Step 1.* Carry out the “EMD” decomposition on the originally collected periodic signal which contains multiple harmonics firstly, and then get a series of component products called “IMF”

*Step 2.* Because the “EMD” process always extracts the most principal information firstly, so the primary “IMF1” component which is firstly got can represent the principal component of the original periodic signal. Then the following fitting calculation will utilize “IMF1.”

*Step 3.* In order to minimize the “end swing” effect caused by EMD as far as possible, with regard to the decomposed IMF1, the sampling data in the middle minizone which have high linearity are selected. We usually utilize only the data sample which have the same number as the sample collected within one cycle of primitive periodic signal, instead of the most sampled data of both ends.

*Step 4.* The sampling sequence in the minizone of the above-mentioned IMF, in accordance with the corresponding moment, is looked on as the sampling sequence within one cycle of a new “periodic signal.”

*Step 5.* This “periodic signal” in single cycle is periodically extended to the both ends; then a continuous “periodic signal” is constructed.

*Step 6.* Select some segment data of the above-mentioned “periodic signal” which can be represented to be  $(C_1, \dots, C_m, \dots, C_M)$  as the sequence of “fitted signal,” and the relationship between this sequence and the sampling sequence of “original signal” which can be represented to be  $(y_1, \dots, y_m, \dots, y_M)$  should be  $C_1 \approx y_1$  and  $c_1 - c_2 \approx y_1 - y_2$  (i.e., initial slope has the same plus or minus sign).

*Step 7.* Apply the “HT” on the “fitted signal” which can be represented to be  $H[c(t)] = c(t) * 1/\pi t = \text{P.V.} \int_{-\infty}^{\infty} (c(t - \tau)/\pi\tau) d\tau$  (where P.V. denotes the Cauchy principal value integral).

*Step 8.* Construct the analytic signal of  $c(t)$ , and the representation can be expressed as  $z(t) = c(t) + iH[c(t)] = a(t)e^{-i\phi(t)}$ . It further demonstrates the local characteristics by polar coordinate expression which is the best local approximation of a trigonometric function, and both amplitude and phase of that trigonometric function change.

*Step 9.* Describe  $c(t)$  in “sine” function form, out of the ordinary, the amplitude and phase of which are changing with time. That well reflects the instantaneous characteristics of data, which is a function with unique amplitude and phase definition that can be expressed as

$$c(t) = a(t) \cos \phi(t) = a(t) \sin \left[ \phi(t) + \frac{\pi}{2} \right]. \quad (22)$$

*Step 10.* The amplitude function  $a(t)$  and the phase function  $\phi(t) + \pi/2$  of this “sinusoidal” can be expressed, respectively, as

$$\begin{aligned} a(t) &= \sqrt{c(t)^2 + H[c(t)]^2}, \\ \phi(t) + \frac{\pi}{2} &= \arctan \left\{ \frac{H[c(t)]}{c(t)} \right\} + \frac{\pi}{2}. \end{aligned} \quad (23)$$

The two representations express the instantaneous amplitude and the instantaneous phase clearly, and they reflect the instantaneous characteristics of data very well.

*Step 11.* Describe the fitted residue  $r_m$  as follows:

$$\begin{aligned} r_m &= y_m - \widehat{C} - c(t_m) \\ &= y_m - \widehat{C} - a(t_m) \sin \left[ \phi(t_m) + \frac{\pi}{2} \right], \end{aligned} \quad (24)$$

where  $y_m$  is the  $m$ th sampling datum,  $\widehat{C}$  is the estimation of the DC component,  $t_m$  is the  $m$ th sampling moment,  $a(t_m)$  and  $\phi(t_m) + \pi/2$  are the instantaneous amplitude and the instantaneous phase of fitted “sinusoidal” at the moment of  $t_m$ , respectively.

*Step 12.* Express the total mean square error by

$$\rho = \sqrt{\frac{1}{M} \sum_{m=1}^M r_m^2}, \quad (25)$$

where  $M$  is the number of elements which sampling sequence contains.

As we know, the ENOBs of ADC can be expressed as  $\text{ENOBs} = N - \log_2(\rho/Q_i)$ , where  $N$  is the number of bits of ADC,  $\rho$  is the total RMS error between the sampling data of originally collected periodic signal containing multiple harmonics and the corresponding points of fitting “sinusoidal”

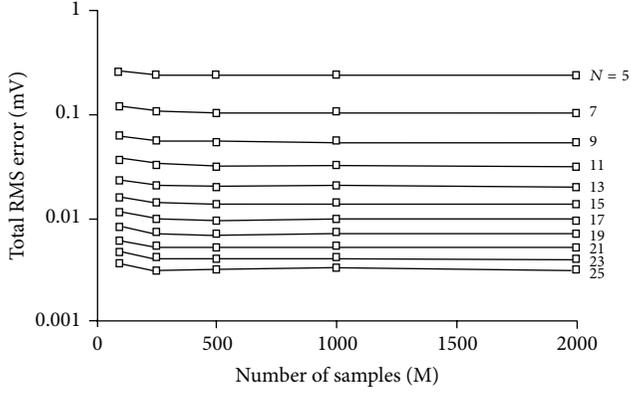


FIGURE 1: The influence of both the highest harmonic number  $N$  of harmonic component and the number  $M$  of sampling points on the total RMS error in the least-squares fitting method for multiple harmonics.

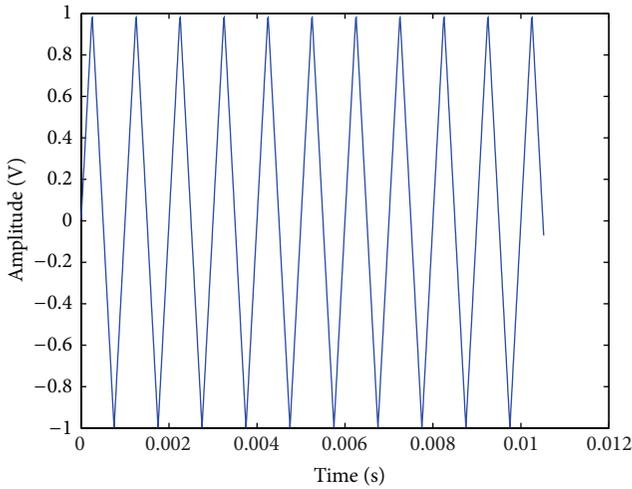


FIGURE 2: The original triangle wave signal source.

wave”, and  $Q_i$  is the theoretical values of RMS quantization error which can be expressed as  $Q_i = E_r / (\sqrt{12} \times 2^N)$ , where  $E_r$  is the range of ADC channel. According to those equations, in the circumstances that both the number of bits of ADC and the range of ADC channel are certain, we can come to a conclusion that the ENOBs of ADC would be evidently larger with the smaller total RMS error.

#### 4. Digital Simulation

Next, to illustrate the effectiveness and the superiority of the new “fitting” method mentioned in this paper, with simulation, we will compare the result of new “fitting” method and that of the least-squares fitting method of multiple harmonics. All parameters are the same as those in [6]: the amplitude of the triangle wave signal source is 1 V, the frequency is 1 kHz, the cycle is 1 ms, and the sampling frequency is 95.08 KHz. That means that 95.08 samples should be sampled in each cycle. The reason to choose that sampling frequency is to

ensure that the phase of signal collected in each cycle was different, which can make the simulation more accord with actual condition and more convincing as well.

As for a triangular wave signal source with the amplitude of  $A$ , the exact harmonic amplitude is

$$D_k = \begin{cases} \frac{8A}{(\pi h)^2} & h \text{ is odd,} \\ 0 & h \text{ is even.} \end{cases} \quad (26)$$

Firstly, the least-squares fitting method for multiple harmonics is taken for simulation. In the fitting process, the total mean square error is the function of the highest harmonic number  $N$  of harmonic component and the number  $M$  of sampling points, as shown in Figure 1. Along with the harmonic number considered in the fitting process becoming higher and higher, the total mean square error comes to be smaller and smaller. Once the number of sampling points considered in the harmonic fitting process is more than 500, then the number of sampling points contributes less on the total RMS error [6].

Then the second approach is taken for simulation, which is based on HHT curve fitting method, which is based on the HHT curve fitting method. Then next, with the acquisition of 10.5 cycles, or in other words, 1000 sampling points of the original signal source being taken as an example, the specific process for application of the HHT method to fit the triangle wave signal source is expounded.

The signal source is shown as Figure 2.

Components of eight IMFs are obtained in proper sequence as the EMD is carried out on the signal source, and the distribution diagrams are shown in Figure 3.

It can be seen from Figure 3 that the first-order IMF component which is decomposed out firstly can reflect the original waveform mostly. We select the sampling high-linearity data in the middle minizone. It should be mentioned out that the data sample must have the same number as the sample collected within one cycle of primitive periodic signal. And so, the single middle full cycle minizone, in accordance with the corresponding time period, can be looked on as the periodic extension base of the periodic wave, and the single cycle waveform is shown as Figure 4.

With periodic extension carried out on the single minizone to both ends, a continuous “periodical signal” is constructed, and with the appropriate interval selected, the “triangular wave” with 1000 sampling points which matches the “original signal” is constructed at last. The constructed waveform is shown as Figure 5.

Thus, the reconstructed “triangular wave” is very similar to the “original signal waveform.” Hence, take the “triangular wave” as the “fitted signal” and apply the “HT” on it, and so the analytical signal is obtained. After the instantaneous amplitude and instantaneous phase of fitted “sinusoidal” are got, the fitted residue can be easily obtained; thus in the end the root mean square error is obtained to be 29.374 mV.

Because of the end effect of EMD, that accuracy is not so ideal and the error is comparatively large. As a result of insufficient sampling points, the “end swing” phenomenon pollutes the whole data sequence, including the medial data.

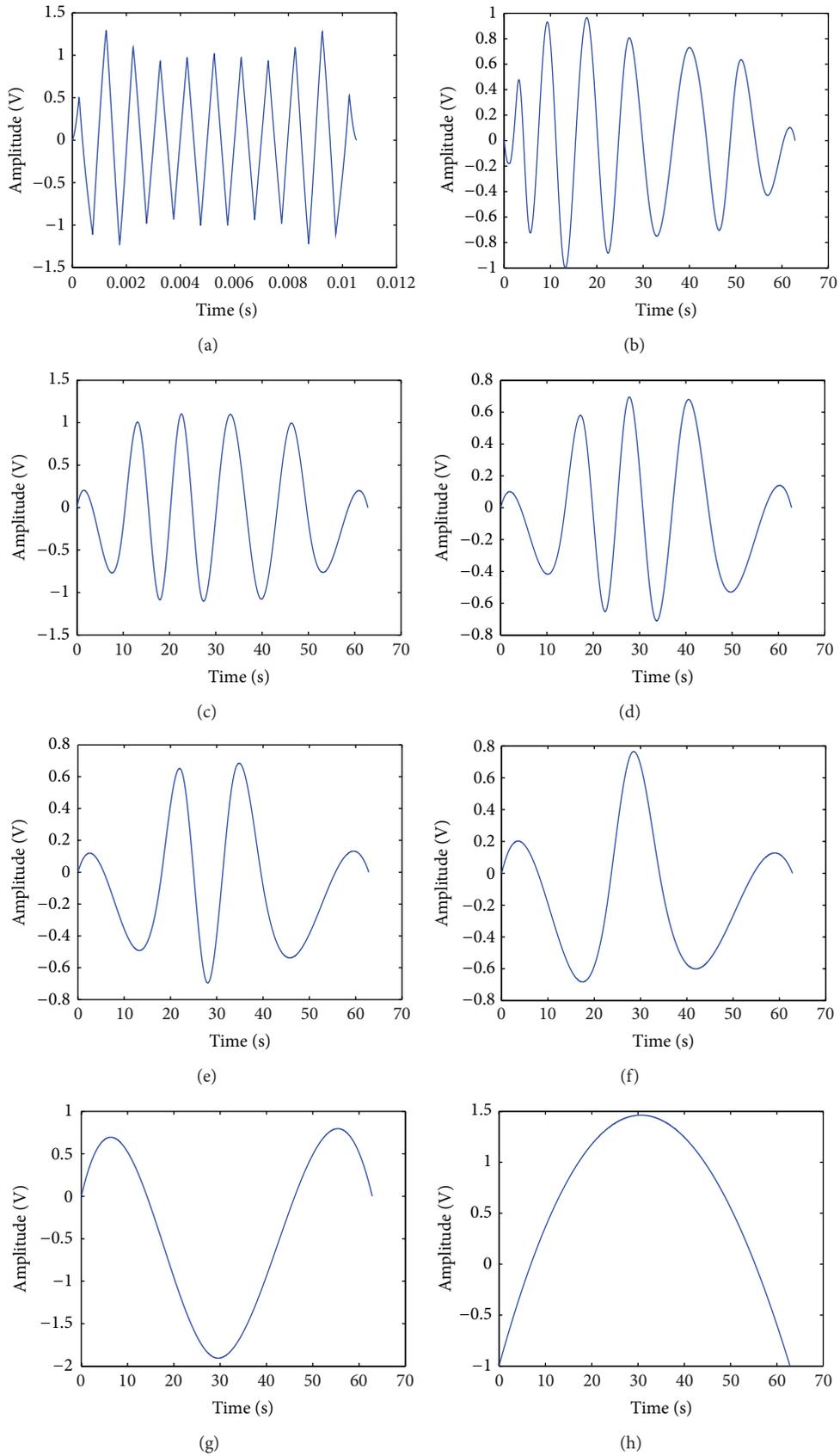


FIGURE 3: The distribution diagrams for components of eight IMFs got with the EMD carried out on the signal source.

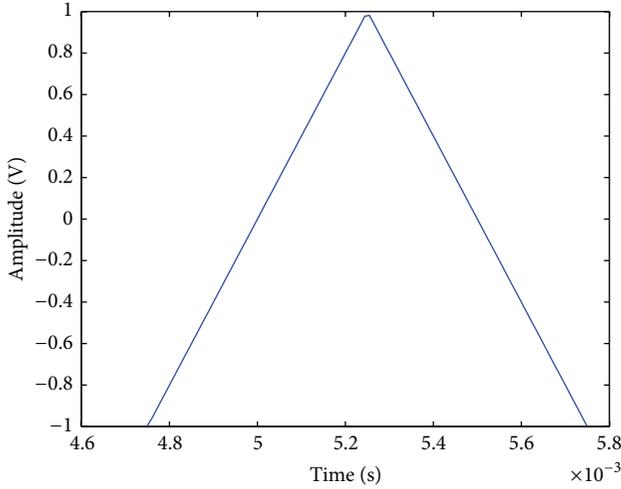


FIGURE 4: The periodic extension base generated on the basis of the first-order IMF component.

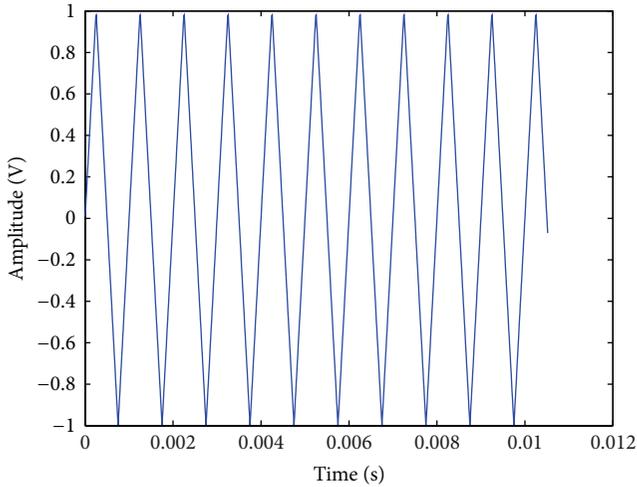


FIGURE 5: The reconstructed “triangle wave” generated after periodic extension.

After the EMD is carried out on the original triangle wave, taking the single middle full “cycle” minizone of IMF1 as the periodic extension base of the fitted triangle wave, it can be found out that the periodic extension base has a large difference from the single cycle of the original triangle wave, and that means there exists relatively large distortion.

Because the end effect is mainly concentrated in both ends of the signal, as the waveform of IMF1 in Figure 3 shows, the end effect has a relatively little effect on the center data. Therefore, in order to decrease the impact of “end swing” to the lowest, we can increase the number of sampling periods so that the number of sampling points is increased; in this way, even if the end effect does exist, it brings rather little contamination on the center data.

Taking simulation experiment of “sinusoidal fitting” for “triangular wave” with sampling points, respectively, of 1500, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000,

TABLE 1: The different RMS error caused by different sampling points after carrying on “sinusoidal fitting” for “triangular wave” based on HHT.

Sampling points	RMS error (mV)
1500	10.571
2000	$4.442 \times 10^{-1}$
3000	$2.282 \times 10^{-2}$
4000	$9.599 \times 10^{-4}$
5000	$8.548 \times 10^{-6}$
6000	$1.415 \times 10^{-7}$
7000	$1.074 \times 10^{-8}$
8000	$3.367 \times 10^{-10}$
9000	$1.735 \times 10^{-12}$
10000	$2.344 \times 10^{-13}$
12000	$2.150 \times 10^{-13}$
20000	$2.312 \times 10^{-13}$

12000, and 20000, the different calculated RMS error can be shown, respectively, in Table 1.

It can be seen from the table that, as the number of sampling points is taken as 1500 or 2000, the RMS error of fitting is not ideal. But once the number is 3000, you can find out that the fitting accuracy can almost compare favourably with the multiple harmonics fitting accuracy with 15 harmonics considered (referring to Figure 1). As the number of sampling points is more than 4000, the fitting precision is even much better than the multiple harmonics fitting precision. For example, as the number of sampling points is taken as 10000, the RMS error goes as far as  $10^{-13}$  mV grade. It can be concluded that the method using the “sinusoidal wave” whose amplitude and phase are both transient to reconstruct the “triangular wave” based on the HHT has almost reproduced the original “triangular wave” perfectly. From the beginning of 10000, even though the number of sampling points increases, the fitting precision could not improve, for that does mainly have something to do with the data accuracy of the computer.

As the “triangular wave” is fitted as above, it can be comprehended by analogy that any periodic wave can be reconstructed by the “sinusoid fitting” algorithm based on the HHT; in other words, any original “periodic wave” can be characterized as the “sinusoidal wave” whose amplitude and phase are both transient.

Unlike the classical least-squares fitting algorithm for multiple harmonic periodic signals which combines “four-parameter algorithm” with “three-parameter algorithm” that presented in the IEEE standards, the new “sinusoid fitting” algorithm based on the HHT avoids selecting the initial values of the parameters and working out the nonlinear equations. So the fact in the IEEE standards that the frequency and amplitude errors from the foremost calculation are propagated to the higher harmonics and the calculation of the  $n$ th harmonic will invariably be contaminated by the errors of the phases and amplitudes of previous steps would be avoided. Firstly, the “EMD” decomposition is carried out

on the original periodic signal source, and the primary firstly-extracted “IMF1” component is got which can most reflect the original signal characteristics. Secondly, the inner data are looked on as the periodic extension base, and then the original waveform is reconstructed. Thirdly, the “HT” is applied on the reconstructed waveform, and then the analytic signal is got. In the end, the reconstructed waveform can be characterized as the “sinusoidal curve” whose amplitude and phase are both transient.

The simulation results show that, once the number of sampling points reaches a certain amount, the reconstructed waveform could be a very good reproduction of the original waveform, and it means that the fitting algorithm based on the HHT method can achieve perfect accuracy.

## 5. Conclusion

Firstly, this paper explains why it would bring great error in the circumstance of nonsinusoidal periodic signal source, if “three-parameter algorithm” or “four-parameter algorithm” that presented in the IEEE standards was merely used to test the ADC, and then the significance of researching multiple harmonics fitting comes out.

Secondly, the principle and realization process of some kind of multiple harmonics fitting method proposed by Portuguese researchers is introduced—that is, the least-square fitting algorithm which combines the “four-parameter algorithm” and “repeated three-parameter algorithm”.

Secondly, the new research method in this paper is introduced in detail—that is, using the “sinusoidal wave” whose amplitude and phase are both transient to reconstruct the “triangular wave” based on the HHT.

That algorithm can be described as follows: the EMD process is carried out on a triangular wave first of all, and then part of the inner data of the originally got first-order IMF is looked on as the periodic extension base to reconstruct the triangular wave. After the “HT” is applied on the reconstructed “triangular wave,” then the corresponding analytic signal is obtained, and so the reconstructed “triangular wave” can be characterized as the special “sinusoidal curve” with parameters changing with time.

Finally, in order to illustrate the effectiveness and the superiority of the new “fitting” method mentioned in this paper, the same example in [6] is given and simulated using our method. The simulation experiment results show that, in the case of enough sampling points, even under the circumstances of low-resolution signal with the harmonic distortion existing, the RMS error between the sampling data of original “triangular wave” and the corresponding points of fitting “sinusoidal wave” is marvelously small. That means the accuracy of new fitting algorithm in this paper is much higher than that in [6].

We can come to a conclusion that the new sine fitting algorithm described in this paper is able to fit a multiharmonic acquired periodic signal. And with this method, the ENOBs of ADC may be tested out, avoiding the trouble of selecting initial value of the parameters and working out the nonlinear equations in the commonly used algorithm.

## Acknowledgments

The authors would like to thank the financial support of the National Natural Science Foundation of China (Grant no. 61104206), the Natural Science Foundation of Anhui Province (Grant no. 1208085QF121), and the Postdoctoral Scientific Research Plan of Jiangsu Province (Grant no. 1102038C). The authors are very grateful to the unknown referees for their valuable remarks.

## References

- [1] A. Song, J. Wu, G. Qin, and W. Huang, “A novel self-decoupled four degree-of-freedom wrist force/torque sensor,” *Measurement*, vol. 40, no. 9-10, pp. 883–891, 2007.
- [2] A. Song, Y. Han, H. Hu, L. Tian, and J. Wu, “Active perception-based haptic texture sensor,” *Sensors and Materials*, vol. 25, no. 1, pp. 1–15, 2013.
- [3] “Standard for digitizing waveform records,” IEEE Std. 1057-1994, 1994.
- [4] “Standard for terminology and test,” *Methods for Analog-to-Digital Converters*, IEEE Std. 1241-2000, 2000.
- [5] R. Pintelon and J. Schoukens, “An improved sine-wave fitting procedure for characterizing data acquisition channels,” *IEEE Transactions on Instrumentation and Measurement*, vol. 45, no. 2, pp. 588–593, 1996.
- [6] P. M. Ramos, M. F. da Silva, R. C. Martins, and A. M. Cruz Serra, “Simulation and experimental results of multiharmonic least-squares fitting algorithms applied to periodic signals,” *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 2, pp. 646–651, 2006.
- [7] J. Schoukens, Y. Rolain, G. Simon, and R. Pintelon, “Fully automated spectral analysis of periodic signals,” *IEEE Transactions on Instrumentation and Measurement*, vol. 52, no. 4, pp. 1021–1024, 2003.
- [8] G. Simon, R. Pintelon, L. Sujbert, and J. Schoukens, “An efficient nonlinear least square multisine fitting algorithm,” in *Proceedings of the 18th IEEE Instrumentation and Measurement of Informatics -Rediscovering Measurement in the Age of Informatics*, pp. 1196–1201, Jun, May 2001.
- [9] T. Andersson and P. Händel, “Multiple-tone estimation by IEEE standard 1057 and the expectation-maximization algorithm,” *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 5, pp. 1833–1839, 2005.
- [10] L. Coluccio, A. Eisinger, and G. Fedele, “A property of the elementary symmetric functions on the frequencies of sinusoidal signals,” *Signal Processing*, vol. 89, no. 5, pp. 765–777, 2009.
- [11] P. Arpaia, A. M. Da Cruz Serra, P. Daponte, and C. L. Monteiro, “A critical note to IEEE 1057-94 standard on hysteretic ADC dynamic testing,” *IEEE Transactions on Instrumentation and Measurement*, vol. 50, no. 4, pp. 941–948, 2001.
- [12] D. L. Carni and G. Fedele, “Multi-Sine Fitting Algorithm enhancement for sinusoidal signal characterization,” *Computer Standards and Interfaces*, vol. 34, pp. 535–540, 2012.
- [13] N. E. Huang, Z. Shen, S. R. Long et al., “The empirical mode decomposition and the Hubert spectrum for nonlinear and non-stationary time series analysis,” *Proceedings of the Royal Society A*, vol. 454, no. 1971, pp. 903–995, 1998.
- [14] N. E. Huang, Z. Shen, and S. R. Long, “A new view of nonlinear water waves: the Hilbert spectrum,” *Annual Review of Fluid Mechanics*, vol. 31, pp. 417–457, 1999.

- [15] R. Yang, A. Song, and B. Xu, "Feature extraction of motor imagery EEG based on wavelet transform and higher-order statistics," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 8, no. 3, pp. 373–384, 2010.
- [16] J. Schoukens, R. Pintelon, and H. Van Hamme, "The interpolated fast Fourier transform: a comparative study," *IEEE Transactions on Instrumentation and Measurement*, vol. 41, no. 2, pp. 226–232, 1992.
- [17] L. Cohen, *Time-Frequency Analysis*, Prentice Hall, Englewood Cliffs, NJ, USA, 1995.

## Research Article

# Evaluation of Circle Diameter by Distributed Tactile Information in Active Tracing

**Hiroyuki Nakamoto, Futoshi Kobayashi, and Fumio Kojima**

*Graduate School of System Informatics, Kobe University, Kobe 657-8501, Japan*

Correspondence should be addressed to Hiroyuki Nakamoto; [hiroyuki.nakamoto@kojimalab.com](mailto:hiroyuki.nakamoto@kojimalab.com)

Received 15 March 2013; Accepted 14 May 2013

Academic Editor: Aiguo Song

Copyright © 2013 Hiroyuki Nakamoto et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Active touch with voluntary movement on the surface of an object is important for human to obtain the local and detailed features on it. In addition, the active touch is considered to enhance the human spatial resolution. In order to improve dexterity performance of multifinger robotic hands, it is necessary to study an active touch method for robotic hands. In this paper, first, we define four requirements of a tactile sensor for active touch and design a distributed tactile sensor model, which can measure a distribution of compressive deformation. Second, we suggest a measurement process with the sensor model, a synthesis method of distributed deformations. In the experiments, a five-finger robotic hand with tactile sensors traces on the surface of cylindrical objects and evaluates the diameters. We confirm that the hand can obtain more information of the diameters by tracing the finger.

## 1. Introduction

Human voluntarily touches an object and perceives its characteristic features. The procedure is called as active touch. Gibson has indicated that the active touch is important for shape perception [1]. In addition, it has been reported that two-position recognition threshold on human finger tip is 2 to 3 mm. The threshold gives us an impression of large value. Then, Loomis and Collins have reported that the sensitivity for stimulus in active touch becomes high [2]. In order to perceive shape of objects, the results indicate an importance of moving hands. Therefore, not only a development of tactile sensor but also a measurement method based on exploratory moving is necessary for developing dexterous robotic hand.

Various studies of robotic hands with tactile sensor have been discussed [3–7]. The tactile sensors are used for recognition. Okamura and Cutkosky have proposed a method which detects small textural features by tracing with robotic fingers [4]. In the case, the hand recognized a local area on an unknown object, and it is called the local exploratory operation. Pribadi et al. have proposed a method to estimate the contour of unknown objects [5]. Allen and Michelman have focused on sensing of global object shapes

and fitting shapes to object models [6]. Wang and Li have applied a tactile sensor for surface tracking of a manipulator [7]. Nakamoto et al. have proposed a method of classifying the shape of an object that a five-finger robot is grasping [8]. The purposes of them are recognitions of unknown area by robotic hands with tactile sensor. Using tactile information, Wang and Li have proposed a feedback to control a robotic arm interacting with soft human skin [9] and have applied an end-effector with a tactile sensor to a massaging work [10]. These studies indicate the importance and effectiveness of robotic hands with tactile sensor. Therefore, the robotic hands are required to increase their applications and become more dexterous.

In order to become more dexterous, the robotic hands should obtain much information of an object. The information is classified roughly into two groups: global rough information and local detail information. The related studies described previously recognize unknown objects and obtain global information. On the other hand, active touch is a method to obtain detailed and local information of objects. If robotic hands can obtain detailed information by active touch as necessary, the information can be used for realizing dexterous operations. The discussions of active touch for

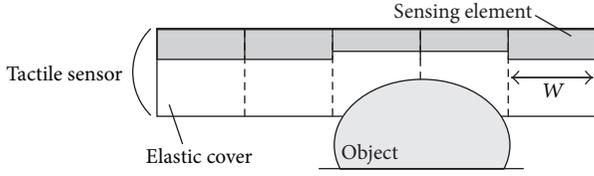


FIGURE 1: Cross-section view of sensor model.

robotic hands are not still enough, and requirements of the tactile sensor which can obtain detailed information by active touch should be also discussed.

In this paper, we aimed at recognizing details of the target area by robotic hand with tactile sensors. The purpose of this study realizes higher spatial resolution than structural spatial resolution of the sensor by tracing the area (hereafter in this paper the tracing operation is referred to as the “active tracing”). In the following sections, we define four requirements of a tactile sensor for active tracing and design a distributed tactile sensor model, which can measure a distribution of compressive deformation. Next, we suggest a measurement process by the sensor model. Then, we confirm the method that can obtain combined distribution to distinguish the circular diameter by robotic hand. Finally, we discuss these results and make a conclusion.

## 2. Active Touch by Distributed Tactile Sensor

**2.1. Sensor Model.** In this section, we describe a modeling of distributed tactile sensor based on knowledge about human skin structure and reveal a mechanism of the high-resolution sensing with active tracing.

Human touches the environment through the skin which covers all of the body. The skin is soft and has a lot of receptors. The receptors detect stimuli to the skin. The skin over the body is one organ which is not being divided along the positions of receptors. Then, we extract the following four features of skin:

- (i) softness;
- (ii) distributed receptors;
- (iii) sheet-like structure;
- (iv) output based on touch area and depth.

A 2D model of distributed tactile sensor based on the first three of four features is shown in Figure 1. The model has a structure which distributes sensor elements and is covered with an elastomer over the elements. The width of one sensor element is  $W$ , and there is no gap between the elements. The seamless elastomer deforms along the shape of touched object. Each sensor element outputs value which responds to the deformation of the upper elastomer. In this model, the width-direction spatial resolution is decided by the width of the sensor element. Therefore, the spatial resolution of the model is  $W$ , and the two-position recognition threshold of the sensor is larger than  $W$ .

**2.2. State Transition in Active Tracing.** When the model touches an object with a constant depth, the contact width between the model and the object is  $L$ . The maximum number of sensor elements, which have outputs based on the contact, is  $\lceil L/W \rceil + 1$ . The width of output distribution is also  $\lceil L/W \rceil + 1$ . Because the elemental width  $W$  is the surplus width about the contact position, the strict position relationship between the object and the element is not determined uniquely from the output distribution. Therefore, even if the number of the output elements  $\lceil L/W \rceil + 1$  and the deformation are constant, the output distribution has changing possibilities within the width  $W$  due to the contact position. In general, the width  $W$  and the differences of the output distribution correspond to the spatial resolution and errors of the sensor, respectively. They are designed and selected according to applications of the sensor. In this study, the active tracing uses the width  $W$  and propose a sensing method which traces the sensor within the width  $W$ .

The contact state transition is shown in the next example. Figure 2 shows an example which traces three times in increments of  $W/4$ . The number of measurement is four. In this time, the object is fixed to the environment, and there is no friction between the object and the sensor.  $t$  means the time step of the trace. The contact width is less than  $2W$  at  $t = 0$ . The object contacts the elastomer of  $e_3$  and  $e_4$  sensor elements. Then, only  $e_3$  and  $e_4$  sensor elements output values of the deformation. At  $t = 1$ , the sensor traces  $W/4$  rightward, the deformation occurs on the elastomer of  $e_2$  element. The deformations of  $e_3$  and  $e_4$  become large and small, respectively. Next, at  $t = 2$ , the deformations of  $e_2$  and  $e_3$  become large and that of  $e_4$  becomes small. Finally, at  $t = 3$ , the deformation of  $e_2$  becomes large, those of  $e_3$  and  $e_4$  become small. In this process, even if the contact width is constant, there are the differences of deformation by tracing smaller width than the spatial resolution.

**2.3. Combination of Sensor Outputs.** In this section, we describe a combination method of sensor outputs in active tracing. The width of one tracing is  $W/N$ , which is based on the element width  $W$ .  $N$  means the number of measurement in one active tracing. In Figure 2, the element  $e_2$  has no deformation at  $t = 0$ , but the deformation becomes large as the tracing progresses. On the other hand, the deformation of the element  $e_4$  becomes progressively smaller. These are continuous changes along the time steps of the trace, and we define the outputs of the sensor as  $e_i(j)$  ( $i = 1, \dots, E$ ,  $j = 0, \dots, N - 1$ ) where  $E$  is the number of the sensor element, and then, we can put them in order of  $(e_i(j))_{j=0,1,\dots,N-1} = e_i(0), e_i(1), \dots, e_i(N - 1)$ . The order is decided by the trace direction.

Next, we focus on the relationship of the elements. The output of element  $e_{i+1}$  at  $t = 0$  corresponds to that of  $e_i$  at  $t = N$ . Assuming that the sensor elements have the same characteristics about output, we can consider that the output  $e_{i+1}(0)$  of element  $e_{i+1}$  at  $t = 0$  and the output  $e_i(N - 1)$  of element  $e_i$  at  $t = N - 1$  are continuous change between the elements by tracing in the width  $W/N$ . The combination of distributed outputs is obtained by putting in order of

TABLE 1: Width of voltage distribution (number of nonzero volt element).

Number of measurement	$\phi 10$	$\phi 15$	$\phi 20$	$\phi 25$	$\phi 30$	$\phi 35$	$\phi 40$
$N = 1$	3	4	4	5	5	5	5
$N = 2$	9	9	9	10	10	11	11
$N = 4$	17	17	18	18	19	20	20
$N = 8$	34	36	37	39	41	42	44

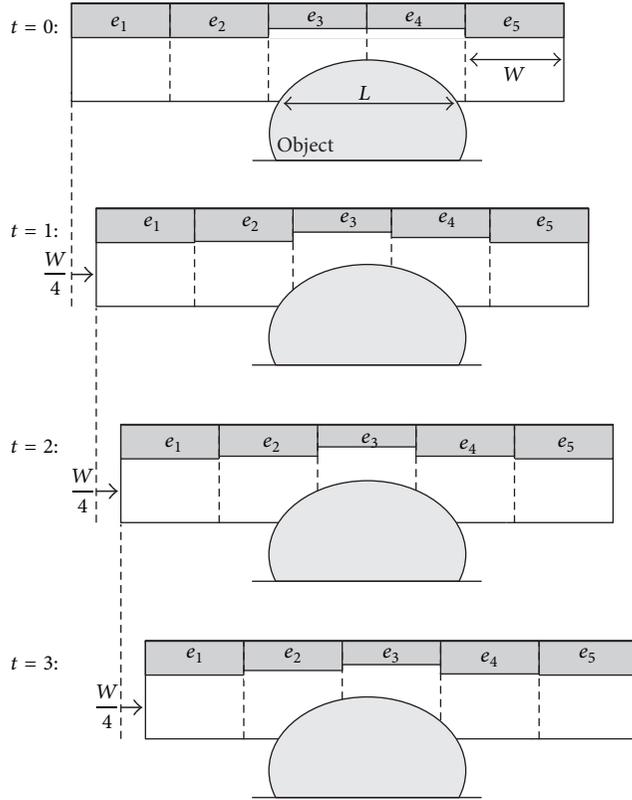


FIGURE 2: Principle of active tracing.

$(e_1(j))_{j=0,1,\dots,N-1}, (e_2(j))_{j=0,1,\dots,N-1}, \dots, (e_E(j))_{j=0,1,\dots,N-1}$ . The combined distribution is the continuous changes by tracing in the width  $W/N$  from  $t = 0$  to  $N - 1$  and includes the details of the contact deformation.

### 3. Tactile Sensor for Robotic Hand

**3.1. Distributed Tactile Sensor.** In order to verify the effectiveness of combined outputs of the distributed tactile sensor in active tracing, we designed and made the sensor as shown in Figure 3. Although the measurement principle of the sensor uses a resistance change at the time of the compressive deformation of the pressure-sensitive conductive rubber as well as the sensor which Shimojo [11], Nakamoto et al. [8], and others proposed as a tactile sensor for a multifinger robot hand, the structure of the sensor is designed in consideration of the features which were described in Section 2.1. The sensor has a three-layer structure. It consists of an elastomer

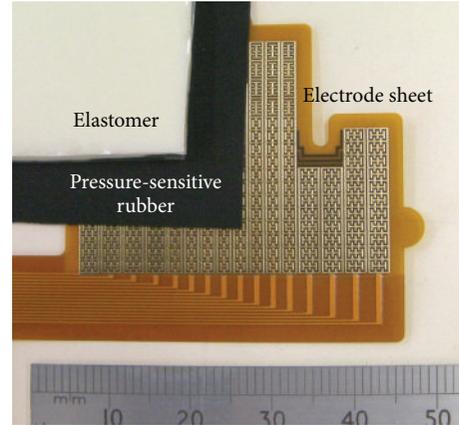


FIGURE 3: Prototype sensor.

sheet, a pressure-sensitive conductive rubber, and an electrode substrate. The elastomer sheet is 2 mm in thickness. If the sheet is too hard, it cannot deform along the contact surface. In contrast, if it is too soft, the pressure on the surface is absorbed by the sheet and the pressure-sensitive rubber cannot deform by the surface pressure. Then, Young's modulus of the elastomer sheet is selected as the same value of the pressure-sensitive rubber. The pressure-sensitive rubber, which is 0.5 mm in thickness, has a characteristic which the electric resistance changes by compressive deformation. The electrode sheet is a flexible substrate. The electrode pattern on the substrate is shown in Figure 4. The electrode is used for the measurement of electric resistance of the pressure-sensitive rubber. An electrode (hereafter is called a unit electrode) is 2 mm square. The size includes 0.2 mm gaps between neighborhood electrodes. Each unit electrode consists of two electrodes for applying voltage and receiving current. The applying-voltage electrodes are printed on the surface of the substrate, and the receiving electrodes are wired inside the substrate via throughholes. The numbers of applying-voltage and receiving electrodes are 16 in the vertical and horizontal directions in Figure 3, respectively. In order to be designed for fitting on the robotic finger, the arrangement of unit electrodes is designed like Figure 3. The number of unit electrodes in the substrate is 204. We also made a multiplexer circuit which selects unit electrodes and amplifies output voltages. A FPGA (Field-Programmable Gate Array) on the same circuit synchronizes selection and measurement. The sampling time of the 204 units is 1 ms.

**3.2. Fundamental Evaluation of Prototype.** We made experiments to evaluate the relationship of outputs of the prototype

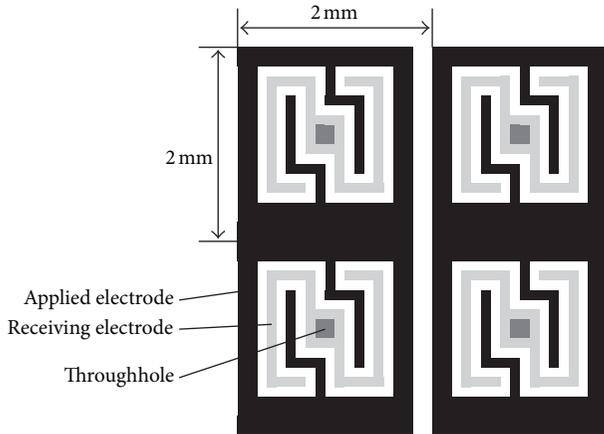


FIGURE 4: Design of electrode.

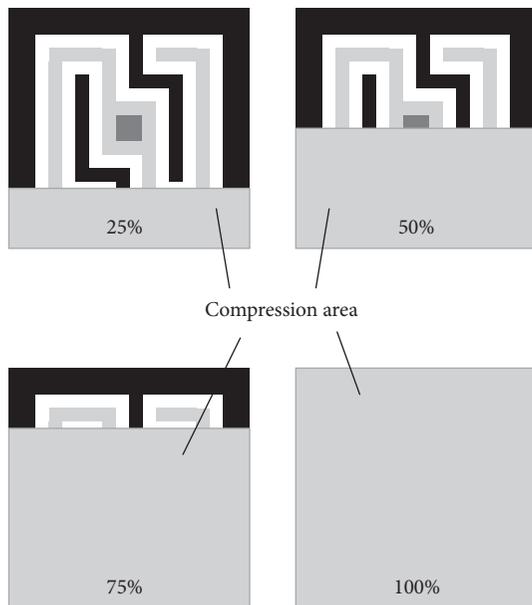


FIGURE 5: Conditions of compression experiment.

sensor to pressured areas. The relationship corresponds to the fourth feature in Section 2.1. In the experiments, we used a 2-axis motorized stage to deform the sensor accurately. The contact object was an acrylic cube. The cube pressed all or partial areas of a unit electrode, and then, we evaluate outputs of the sensor. Then, we decided four different contact conditions that the pressed areas of a unit electrode are 100, 75, 50, or 25% as shown in Figure 5. The cubic surface was set accurately parallel to the sensor surface, and the motorized stages move the cube in the direction of compression. The range of the movement was from 0 to 1.0 mm, and the length of one movement was 0.1 mm.

The experimental results are shown in Figure 6. The horizontal and vertical axes in Figure 6 are the press depth and output voltage, respectively. The results show that the output voltage became high by a constant slope as the amount of compression becomes large from 0 to 1.0 mm. The ratios

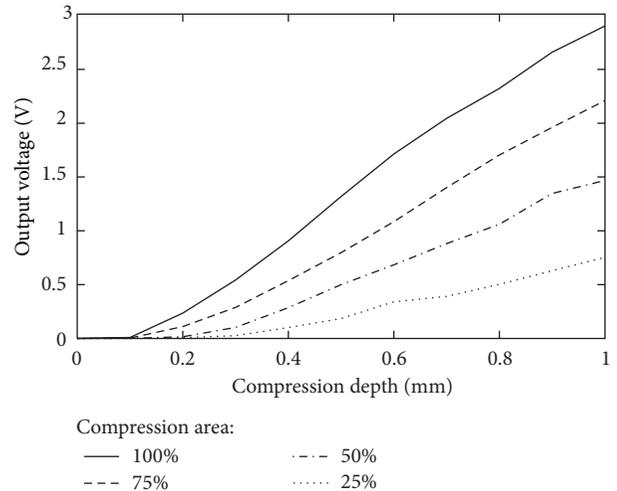


FIGURE 6: Compression depth versus output voltage relation.

of compression area correspond to the output voltages in especially 1.0 mm. For example, about 4 times of the output voltage of 25% area were mostly in agreement with the output voltage of 100% area. However, when compressive deformations were small, there were also the results that the output voltages had not changed. For example, the output voltage of 50% area increased after 0.2 mm. This cause is considered that small compressive deformations were absorbed by the surface elastomer sheet. As described previously, the contact area of compressive deformation is proportional to the output voltage in the condition of a certain amount of press depth. An equation of the relation is defined later by the least mean square approximation,

$$v_o = 3.45pq - 0.39, \quad (1)$$

where  $v_o$  is the output voltage,  $p$  ( $0 \leq p \leq 1.0$ ) and  $q$  are the contact area and the compression depth, respectively. Since the amount of deformation equals the product of the contact area and the compression depth, the output voltage is proportional to the amount of deformation. Furthermore, if the deformation of the surface is a partial area, the output voltage increases in response to the area. Therefore, we confirmed that the prototype has the fourth function described in Section 2.1.

## 4. Experiments with Active Tracing

**4.1. Multifinger Robotic Hand.** A photograph of multifinger robotic hand and the arrangement of its DOFs are shown in Figure 7 [12]. The robotic hand has five fingers and a total of 16 degrees of freedom and has tactile sensors shown in Figure 7 on its finger pads. Although the tactile sensor also covers finger tip and side, the finger contacts only with the finger pad in the next experiments. The length between the middle finger tip and the palm bottom is 290 mm. The hand is controlled by a PC with RT Linux as the operating system. It can precisely be operated by the PD control.

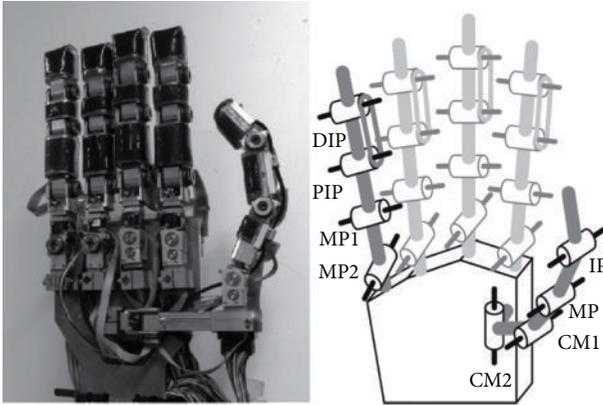


FIGURE 7: Multifinger robotic hand.

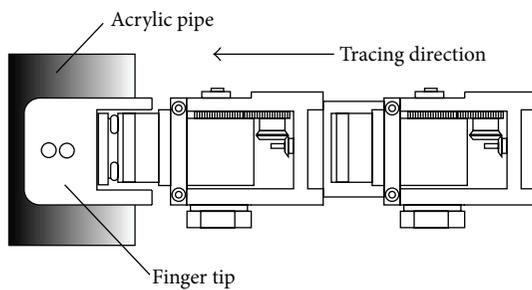


FIGURE 8: Tracing direction.

**4.2. Combination of Output Voltage.** We used seven acrylic pipes as objects. The robotic hand touches and traces on them in experiments. The pipes are 10, 15, 20, 25, 30, 35, and 40 mm in diameter. The lengths of the pipes are 20 mm. First, the pipe is fixed on the table. Next, the finger presses the pipe 0.7 mm in the normal direction to the pipe. Then, the finger traces on the pipe in the direction perpendicular to the shaft of the pipes and the direction parallel to the surface of the sensor. The finger tip is equipped with a 3-axis force sensor and kept contact force constant. Because the unit electrode is 2-mm-square, we decided 1.0, 0.5, and 0.25 mm as the lengths of one tracing. In response, the finger traced and measured 2, 4, and 8 times, respectively. The tracing direction is shown in Figure 8. The finger can trace on the pipes at a constant velocity and measured output distributions at each tracing length. After measurements of each 2, 4, and 8 times were completed, we obtained combined distributions from the measured outputs by the order described in Section 2.3. Because outputs in the direction of the shaft do not change by using the pipes as objects, we extracted only output distributions of a pipe-radial direction that had the highest output value. In order to compare results, we also obtained outputs without tracing.

The results are shown in Figure 9. In order to compare easily, outputs of zero volt part except for both ends of the distribution are not shown in Figure 9.  $N$  expresses the number of measurement in one active tracing as described in Section 2.3. In the graphs, the horizontal axes mean the sequential element number of combined voltage distribution,

and the vertical axes mean the output voltage of each element. Figure 9(a) shows the results of one measurement without tracing, and the output distributions are most of the same shapes and the widths even if the diameters of pipe are different. The widths of valued distribution are from 3 to 5, and they are not useful to distinguish the pipe diameters. It is difficult to estimate the diameter of the pipe from an output of single measurement. On the other hand, the combined outputs in  $N = 2$  and  $N = 4$  measurements have double and quadruple widths of single-measurement distributions, respectively. The width of output distribution tends to increase as the pipe diameter becomes large. The distribution widths are from 34 to 44 in  $N = 8$  and depend on the pipe diameters.

Next, we focus on the shapes of output distribution. The shapes of distribution in  $N = 2$  are almost the same shapes of distribution expanded to double single-measurement outputs in the width direction. In contrast, the combined distributions in  $N = 4$  have some high values and peaks of arc shape. There are differences between the shapes of  $N = 4$  and those of the quadruple distribution of single measurement even though the pipe diameters are the same. Then, the shapes of distribution have difference by increasing the number of measurement. The results show that the combined distribution has details of the shape by active tracing. According to Figure 9(d) which includes the results in  $N = 8$ , the distribution widths of each diameter have difference. The loose peaks of distribution are obtained by tracing the finger in  $N = 4$  and  $N = 8$ . We extend our investigations of the distribution widths and the loose peaks.

We extracted feature quantities from the combined distributions. Table 1 shows the widths of the combined distributions. The widths mean the numbers of element of distribution not including 0 volt. Table 2 shows the averages of absolute difference between neighborhood elements. Each distribution width has correspondence with value which is obtained by multiplying the number of measurement by the width in  $N = 1$ . Then, the variation of distribution widths is 3 under the conditions of  $N = 1$  and  $N = 2$ . In  $N = 4$ , the variation is 4 and is 7 at  $N = 8$ . The results in  $N = 8$  show that the finger can classify diameters of pipe by the width of combined distribution and some thresholds. The combined distributions reflect small change of compressive deformation and give high spatial resolution to the sensor. On the other hand, we can confirm the resolution in the height direction of distribution in Table 2. The averages become smaller along with the increasing of the number of measurement. Then, the distributions in Figure 9(d) have gentler curves than those in Figure 9(a). Furthermore, the large diameter pipes have the small averages. For example, in  $N = 4$  or  $N = 8$ , each average becomes small gradually as the pipe diameters become large. A large diameter pipe has a gentle curve on a touch surface, and the averages reflect the details despite the small deformations. The increasing of the number of measurement strengthens the tendency to that. Therefore, we confirmed that the spatial resolution of the sensor becomes high by the tracing measurement.

In order to confirm the spatial resolution for pipe diameter classification with the sensor tracing, we focused on

TABLE 2: Average of difference between neighborhood members of voltage distribution.

Number of measurement	$\phi 10$	$\phi 15$	$\phi 20$	$\phi 25$	$\phi 30$	$\phi 35$	$\phi 40$
$N = 1$	0.522	0.481	0.436	0.413	0.485	0.454	0.418
$N = 2$	0.321	0.320	0.299	0.309	0.293	0.289	0.275
$N = 4$	0.181	0.177	0.169	0.159	0.140	0.141	0.140
$N = 8$	0.103	0.097	0.095	0.089	0.089	0.083	0.079

(The unit is V.)

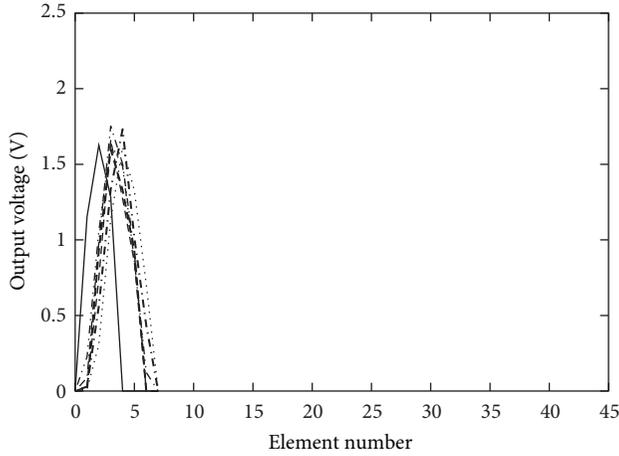
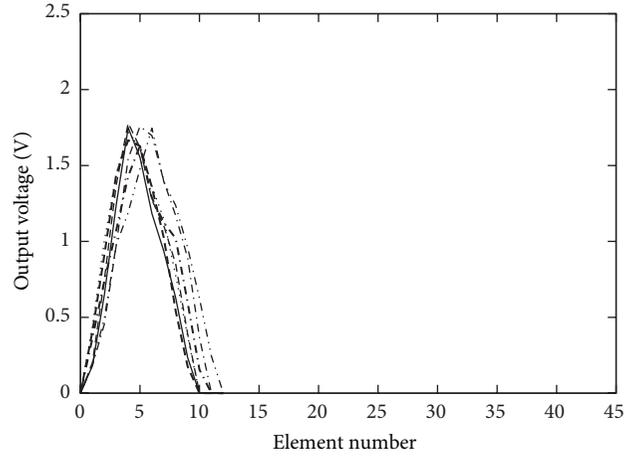
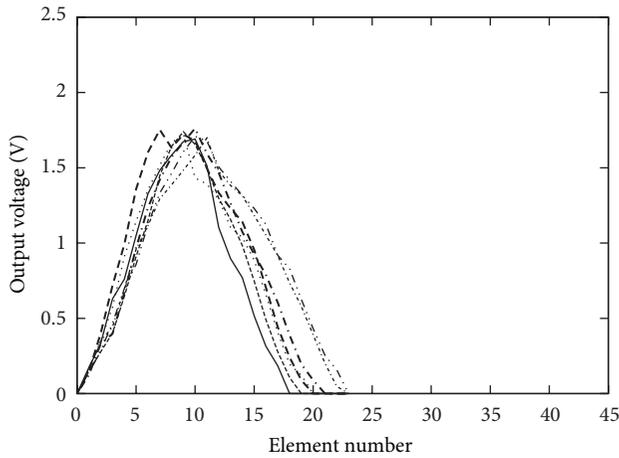
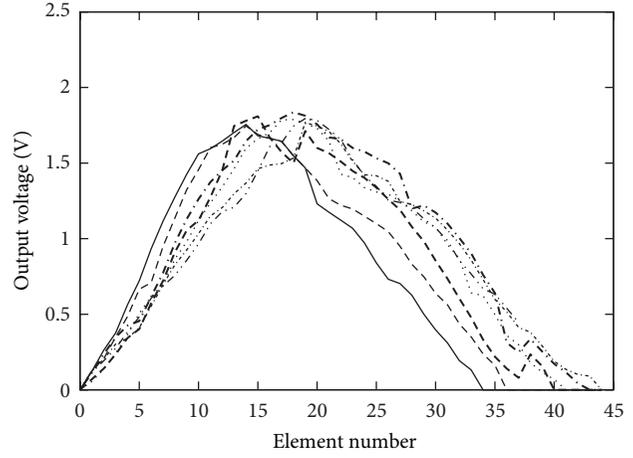
(a)  $N = 1$ (b)  $N = 2$ (c)  $N = 4$ (d)  $N = 8$ 

FIGURE 9: Results of combined voltage distribution.

the classification by the widths of combined distribution. In Table 1, the distributions in  $N = 1$  and  $N = 2$  have three widths. However, we described in Section 2.2 that a width of combined distribution is not constant because of effects of relative position between the sensor and the object. For example, in the case of  $N = 1$ , the distribution width of 25 mm pipe in diameter is 4 or 5. Through repetitive

experiments, the single measurement could distinguish 10 or 40 mm in diameter by the distribution width without depending on the relative position. The result means that the spatial resolution of the sensor was 30 mm in diameter. On the other hand, the measurements of 4 times had the spatial resolution of 15 mm in diameter. The resolution of the measurement of 8 times was 10 mm in diameter. These results

show that the robotic hand can distinguish 10 mm in pipe diameter. We confirmed that the robotic hand can distinguish the diameters at the high resolution by tracing the finger.

## 5. Conclusion

In this paper, in order to obtain details of local area of an object, we decided the conditions of distributed tactile sensors and verified the effectiveness of tracing the finger with the tactile sensor. As the results of the experiment by the robotic finger, the finger could obtain combined distributions and distinguish pipe diameters by 10 mm. Although we have used 1D distributions for combined distributions in this paper, the tactile sensors on the robotic hand can output 2D voltage distributions. In order to obtain combined 2D distributions, we are extending the proposed method and are trying to obtain 3D touch surfaces from combined 2D distributions at the next step of this study.

The sensor can obtain details of local area by the proposed method. Although the miniaturization of the tactile sensor is required in order to improve the accuracy of touch perception, we also consider that it is necessary to combine the proposed local area measurement and a global area measurement [13]. To increase efficiency of the measurement, a high-speed performance is required of the global area measurement. We will apply the sensor to a robotic hand and propose a measurement method of whole shape of an object by the combination of the local and global area measurement.

## References

- [1] J. J. Gibson, "Observations on active touch," *Psychological Review*, vol. 69, no. 6, pp. 477–491, 1962.
- [2] J. M. Loomis and C. C. Collins, "Sensitivity to shifts of a point stimulus: an instance of tactile hyperacuity," *Perception and Psychophysics*, vol. 24, no. 6, pp. 487–492, 1978.
- [3] T. Mouri, H. Kawasaki, and K. Umehayashi, "Developments of new anthropomorphic robot hand and its master slave system," in *Proceedings of the IEEE IRS/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 3474–3479, August 2005.
- [4] A. M. Okamura and M. R. Cutkosky, "Feature detection for haptic exploration with robotic fingers," *International Journal of Robotics Research*, vol. 20, no. 12, pp. 925–938, 2001.
- [5] K. Pribadi, J. S. Bay, and H. Hemami, "Exploration and dynamic shape estimation by a robotic probe," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 4, pp. 840–846, 1989.
- [6] P. K. Allen and P. Michelman, "Acquisition and interpretation of 3-D sensor data from touch," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 4, pp. 397–404, 1990.
- [7] J. Wang and Y. Li, "Surface-tracking of a 5-DOF manipulator equipped with tactile sensors," in *Proceedings of the 11th International Conference on Control, Automation, Robotics and Vision (ICARCV '10)*, pp. 2448–2453, Singapore, December 2010.
- [8] H. Nakamoto, F. Kobayashi, N. Imamura, H. Shirasawa, and F. Kojima, "Shape classification in rotation manipulation by universal robot hand," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '08)*, pp. 53–58, Nice, France, September 2008.
- [9] J. Wang and Y. Li, "A study on a robotic arm contacting with human skin using tactile sensing feedback strategies," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '10)*, pp. 1559–1564, Tianjin, China, December 2010.
- [10] J. Wang and Y. Li, "Massaging human feet by a redundant manipulator equipped with a tactile sensor," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '10)*, pp. 7–12, Montreal, Canada, July 2010.
- [11] M. Shimojo, "Mechanical filtering effect of elastic cover for tactile sensor," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 1, pp. 128–132, 1997.
- [12] W. Fukui, F. Kobayashi, F. Kojima et al., "Fingertip force and position control using force sensor and tactile sensor for universal robot hand II," in *Proceedings of the Workshop on Robotic Intelligence in Informationally Structured Space (RIISS '11)*, pp. 43–48, Paris, France, April 2011.
- [13] R. L. Klatzky and S. J. Lederman, "Stages of manual exploration in haptic object identification," *Perception and Psychophysics*, vol. 52, no. 6, pp. 661–670, 1992.

## Research Article

# Force-Sensor-Based Estimation of Needle Tip Deflection in Brachytherapy

Thomas Lehmann,<sup>1</sup> Mahdi Tavakoli,<sup>1</sup> Nawaid Usmani,<sup>2</sup> and Ronald Sloboda<sup>2</sup>

<sup>1</sup> Department of Electrical and Computer Engineering, University of Alberta, 9107-116 Street, Edmonton, AB, Canada T6G 2V4

<sup>2</sup> Cross Cancer Institute, University of Alberta, 11560 University Avenue, Edmonton, AB, Canada T6G 1Z2

Correspondence should be addressed to Thomas Lehmann; lehmann@ualberta.ca

Received 15 March 2013; Accepted 4 May 2013

Academic Editor: Aiguo Song

Copyright © 2013 Thomas Lehmann et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A virtual sensor is developed for the online estimation of needle tip deflection during permanent interstitial brachytherapy needle insertion. Permanent interstitial brachytherapy is an effective, minimally invasive, and patient friendly cancer treatment procedure. The deflection of the needles used in the procedure, however, undermines the treatment efficiency and, therefore, needs to be minimized. Any feedback control technique to minimize the needle deflection will require feedback of this quantity, which is not easy to provide. The proposed virtual sensor for needle deflection incorporates a force/torque sensor, mounted at the base of the needle that always remains outside the patient. The measured forces/torques are used by a mathematical model, developed based on mechanical needle properties. The resulting estimation of tip deflection in real time during needle insertion is the main contribution of this paper. The proposed approach solely relies on the measured forces and torques without a need for any other invasive/noninvasive sensing devices. A few mechanical models have been introduced previously regarding the way the forces are composed along the needle during insertion; we will compare our model to those approaches in terms of accuracy. In order to conduct experiments to verify the deflection model, a custom-built, 2-DOF robotic system for needle insertion is developed and discussed. This system is a prototype of an intelligent, hand-held surgical assistant tool that incorporates the virtual sensor proposed in this paper.

## 1. Introduction

Permanent interstitial brachytherapy is a cancer treatment procedure, in which radioactive seeds are implanted in tissue (e.g., the prostate, see Figure 1) in order to eliminate the cancer from inside. This procedure has emerged as an effective, minimally invasive, patient friendly, and cost-effective treatment option. For maximum treatment efficiency, the seeds have to be placed at exact locations inside and around the tumor that are determined in the preoperative planning stage. The radioactive seeds are initially loaded inside special needles. Intraoperatively, the seed-carrying needles are manually advanced toward planned locations where the seeds are deposited.

Two critical assumptions in the above procedure are that the needles will remain parallel across the entire length of their insertion in tissue and that the tissue will not deform

as the needles penetrate it. However, in practice, neither assumption holds well, causing the actual needle trajectories to not pass through the planned locations. In fact, current manual needle insertion techniques for prostate treatment can place seeds with an accuracy of only 5 mm, which is a substantial error given the average prostate size. As a result, due to delivery of a different radiation dose, the radiation might not have the desired effect on the cancerous tissue and could instead undermine healthy tissue.

The predominant causes of inaccuracy in seed placement are needle deflection and tissue deformation during needle insertion/retraction [1, 2]. In general, needle bending is a function of the needle geometry [3, 4]. For instance, needles experience more bending with smaller diameters and with beveled tips. Beveled tips (Figure 2) are needed for easily cutting into tissue [5]. With a beveled tip, needle bending is larger for smaller bevel angles [6].



FIGURE 1: View of a prostate brachytherapy procedure as it is currently performed.

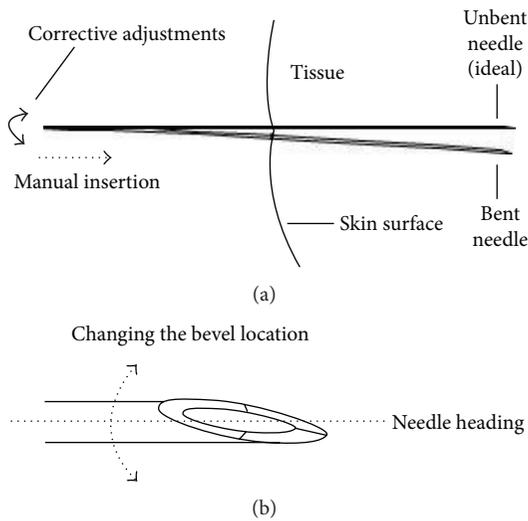


FIGURE 2: (a) A needle's bevel tip causes it to have a curved trajectory in tissue. (b) The needle's bevel location, which is the angular position of the bevel relative to the needle axis, can be adjusted to steer the needle.

Needle deflection and tissue deformation are coupled effects and influence each other. For simplicity, past studies have tried to decouple these effects by considering one of two extreme cases: a rigid needle in deformable tissue or a flexible needle in rigid tissue. For the second case, a bicycle-like kinematic model predicting the needle tip position was proposed [7]. A result pertinent to this paper is that when a beveled-tip flexible needle is pushed through tissue, the asymmetry of the tip causes the needle to bend (see Figure 2(a)). This is intuitively understood because the asymmetric geometry of the beveled tip leads to a force on the tip in the same direction as the bevel and a displacement of tissue to that side [5].

To improve needle targeting accuracy despite the bevel effect, there is active research on high-resolution imaging, tissue modeling, and robotic insertion preplanning. However, the resulting systems are still too complex and costly for the operating room. There is currently a need for a simple, inexpensive, and effective instrument in the surgeons' hands that can enable them to place the seeds more accurately.

We are working toward a low-cost and effective hand-held instrument for use in prostate needle insertion that automatically corrects the needle tip's bevel location during manual insertion in order to improve needle targeting and seed placement accuracy; the bevel location is defined as the angular position of the bevel tip relative to the needle's longitudinal axis (Figure 2(b)). A main idea related to needle steering is to leverage the very tendency of the bevel-tip needle for traveling on a curve to bring the needle tip back on the intended path once the needle has been bent, regardless of what caused the needle deflection [8]. To this end, the needle's bevel location can be changed by a feedback control algorithm based on the *real time feedback of needle deflection*. The feedback control algorithm will steer the needle into the opposite direction when needle deflection exceeds a set threshold by simply turning the needle about its longitudinal axis by  $180^\circ$ . This causes the forces acting at the needle tip to point in the opposite direction and steer the needle back toward the unbent ideal path. Since the surgeon is fully in charge of inserting the needle and the computerized system only modifies the bevel location, the hand-held instrument will be intrinsically safe to use.

The contribution of this paper lies in real time measurement of needle deflection using our proposed novel *virtual sensor*; incorporating a *physical sensor* for measuring this quantity is next to impossible in a clinical setting at least with the current technology. To measure the needle deflection during insertion into soft tissue, one may consider using 3D ultrasound imaging. However, the detection of the typical needle tip deflections in the order of several millimeters will be difficult due to the low resolution of ultrasound images and the associated high computational demands, notwithstanding the high costs of integrating a 3D imager and the required image processors in a computerized needle insertion system. Using 2D ultrasound imaging is not suitable because deflection measurement requires that the needle be always visible in the image plane while in practice the surgeon needs to intermittently switch between various sagittal and axial planes to monitor the insertion progress. Another possibility to measure the needle deflection is to use an electromagnetic tracking system, such as the Aurora Electromagnetic Tracking System by NDI, which involves a sensor embedded in the needle. However, this is only appropriate for in vitro testing where invasiveness is not an issue; sterilization requirements and the preloading of seeds inside the needle make in vivo utilization of a tracker inside the needle prohibitive. Therefore, in this paper, we pursue a noninvasive virtual sensing approach to needle deflection measurement in soft-tissue needle insertion. The foundation of our proposed virtual sensor is the hypothesis that the transverse force and the bending moment at the needle base predict the needle's longitudinal deflection inside the tissue. Therefore, in this paper, as the needle is being inserted into tissue, the forces and moments reflected at the needle base are measured by a physical sensor. Then, a needle tip deflection model and virtual sensor are developed solely based on this force/moment at the needle base. This virtual needle deflection sensor can be applied to a wide range of

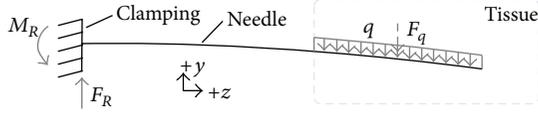


FIGURE 3: The distributed load  $q$  acting along the inserted portion of the needle as a reaction force to the needle, as proposed by Kataoka et al. [3].  $F_R$  is the resulting force at the needle base. The 2D diagram plane is the deflection plane of the needle. The dashed arrow represents the resulting force acting at the geometric center (centroid) of the area of the distributed load  $q$ .

percutaneous medical procedures that involve needles prone to deflection.

## 2. Methods and Materials

**2.1. The Needle Deflection Model.** The developed needle model is fundamentally based on a static model for deflection of a beam. The needle can be regarded as a cantilever beam since its only fixation is at the base. The equations are based on the Euler-Bernoulli beam theory, which relates the loads or forces applied to the beam to its deflection. As the needle deflects increasingly during insertion in soft tissue, it acts as a loaded spring that tries to return to its initial unbent state but is kept in place by the tissue. The needle exerts a distributed load perpendicular to the needle axis onto the tissue. To keep the needle in its bent state, that is, to maintain the equilibrium condition, the tissue in return reacts with a distributed load along the needle (Figure 3). In general, distributed loads can be replaced by a resultant point force (concentrated load) that acts at a specific point along the needle. This reduction allows for a simplification of the equations used to calculate the deflection. The location of the resultant force is the *geometric center* or *centroid* of the area of the distributed load [9]. The magnitude of the force acting on the needle base ( $F_R$ ) is the integral of the load and can be expressed as

$$F_R = F_q = \int_{L-l_0}^L q(z) dz \quad (1)$$

or

$$F_R = F_q = q \cdot l \quad (2)$$

for an equally distributed load, where  $q$  is the load per unit length,  $l_0$  represents length portion outside of tissue, and  $l$  is the length portion of the needle, which is inserted into tissue. It should be noted that the indicated force at the needle clamping is the resistance force exerted by the clamp onto the needle. While the measured force ( $F_S$ ) is equal in magnitude to  $F_R$ , the directions are opposed. The relation between  $F_R$  and  $F_S$  can be expressed as

$$F_R = -F_S. \quad (3)$$

Since the only support of the needle is at its base, the full magnitude of the resulting force and moment as well as the direction are reflected at the needle base and can be

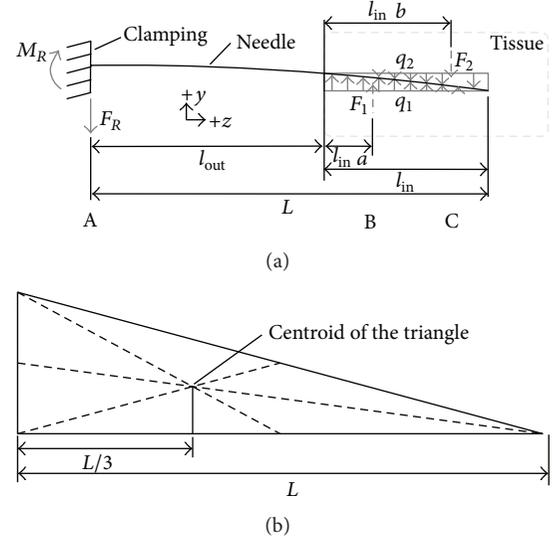


FIGURE 4: (a) The triangularly distributed loads  $q_1$  and  $q_2$  acting along the needle while inserting into tissue.  $F_R$  is the resulting force at the needle base.  $F_1$  and  $F_2$ —shown with dashed lines—are the resulting forces from the distributed loads  $q_1$  and  $q_2$ . (b) The location of the centroid of a triangle.

measured. This means that the expected force measurement results should have the same direction as the deflection. Insertion experiments, however, showed a force at the needle base in opposite direction. Figure 8(a) shows forces and torques for a deflection in negative  $y$ -direction. This means that the forces exerted by the tissue cannot only be one distributed load acting in negative  $y$ -direction (cf. Figure 3) but must be comprised of multiple loads acting in positive and negative  $y$ -direction. Tissue below the needle can be considered as acting as a support, and tissue above the needle still exerts a distributed reaction force on the needle.

The triangular distribution of  $q_1$  in Figure 4(a) can be explained by assuming that tissue is pushed down more towards the side where the needle enters the tissue. As the needle penetrates deeper into tissue, the deflection increases, which leads to pressure being applied on the tissue in negative  $y$ -direction. Tissue, which is horizontally closer to the point of insertion (here: left side, see Figure 4(a)), is exposed to this pressure for an increased period. Therefore, the distribution of the load applied on the needle by tissue is at its maximum at the insertion point and decreases in  $z$ -direction. This assumption was also made by Abolhassani et al. [10]. They however did not include it in their model, as it had very little impact on their case. For the load above the needle, a reverse effect applies. As the tissue closer to the needle entry point is pushed towards the direction of deflection increasingly, the tissue above the needle in this area cannot apply any resistance force from above.

As before mentioned, the loads  $q_1$  and  $q_2$  can be reduced to the concentrated loads  $F_1$  and  $F_2$ , which act at the centroid of the triangularly distributed load. The centroid of a triangle and its location is illustrated in Figure 4(b). The figure shows that the centroid of a triangularly distributed load is at  $L/3$ ,

where  $L$  is the length of the triangle's leg. Thus, the factors  $a$  and  $b$  should have the values  $1/3$  and  $2/3$ , respectively (see Figure 4(a)). It should be noted that the triangular distribution of loads  $q_1$  and  $q_2$  is an assumption.

A common method to finding deflections in beam structures is twice integrating the bending moment equation:

$$M = EI \frac{d^2 v}{dz^2}, \quad (4)$$

where  $M$  and  $v$  are the moment and deflection at a distance  $z$  from the base and  $E$  and  $I$  are the Young's modulus of stainless steel (200 GPa) and the area moment of inertia of the needle. The equation for the moment  $M$  can be found by analysis of the free body diagram. In our case there are multiple forces acting simultaneously ( $F_1$  and  $F_2$ ) at different locations. Therefore, the method of superposition needs to be applied. This method allows for regarding multiple forces acting separately on the beam and superimposing the deflections resulting from each force [11]. The deflections for each force are

$$\begin{aligned} \delta_1 &= \frac{F_1(l_{\text{out}} + l_{\text{in}}a)^2 (3L - l_{\text{out}} - l_{\text{in}}a)}{6EI}, \\ \delta_2 &= -\frac{F_2(l_{\text{out}} + l_{\text{in}}b)^2 (3L - l_{\text{out}} - l_{\text{in}}b)}{6EI}, \end{aligned} \quad (5)$$

([11, p. 1084]) where  $\delta_1$  and  $\delta_2$  are the deflections at the needle tip for  $F_1$  and  $F_2$ , respectively,  $l_{\text{in}}$  and  $l_{\text{out}}$  are the length proportions inside and outside of tissue,  $L$  represents the total needle length, and  $a$  and  $b$  are factors to adjust the points at which forces  $F_1$  and  $F_2$  act. Superimposing the two deflections results in

$$\begin{aligned} \delta_{1,2} &= \frac{F_1(l_{\text{out}} + l_{\text{in}}a)^2 (3L - l_{\text{out}} - l_{\text{in}}a)}{6EI} \\ &\quad - \frac{F_2(l_{\text{out}} + l_{\text{in}}b)^2 (3L - l_{\text{out}} - l_{\text{in}}b)}{6EI}, \end{aligned} \quad (6)$$

where  $\delta_{1,2}$  is the deflection at the needle tip for forces  $F_1$  and  $F_2$  combined. The up to now unknown forces  $F_1$  and  $F_2$ , with which a relation to  $F_R$  and  $M_R$  can be established, can be obtained from equilibrium conditions. These equilibrium conditions apply in the force and moment system of the free body diagrams in Figure 5. The cut in the free body diagram in Figure 5(a) was chosen because, in this case, the force  $F_1$  will cancel out at both sides, as  $F_1$  acts in the same direction at both sides of the cut. Force  $F_1$  at point B can also be regarded as support, which means that no shear forces are acting at the cut. Since in every point during the insertion equilibrium conditions are assumed, the moments acting at both the left and right sides of B must sum up to zero:

$$\begin{aligned} \text{left side: } \sum M &= -M_R - M + F_R(l_{\text{out}} + l_{\text{in}}a) = 0 \\ \iff M &= -M_R + F_R(l_{\text{out}} + l_{\text{in}}a), \end{aligned} \quad (7)$$

$$\begin{aligned} \text{right side: } \sum M &= M - F_2 l_{\text{in}}(b - a) = 0 \\ \iff M &= F_2 l_{\text{in}}(b - a). \end{aligned}$$

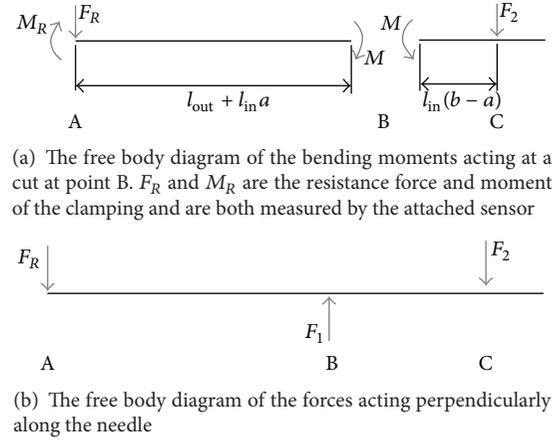


FIGURE 5: Free body diagrams of the needle while inserting into tissue.

As the bending moments at each side of the cut must be equal, (7) can be equalized to

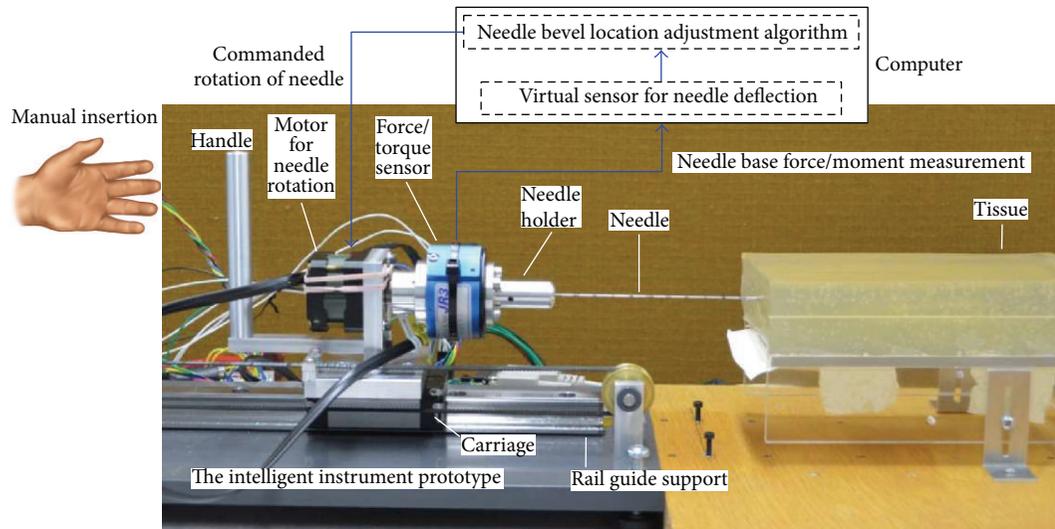
$$\begin{aligned} -M_R + F_R(l_{\text{out}} + l_{\text{in}}a) &= F_2 l_{\text{in}}(b - a) \\ \iff F_2 &= \frac{-M_R + F_R(l_{\text{out}} + l_{\text{in}}a)}{l_{\text{in}}(b - a)}, \end{aligned} \quad (8)$$

and  $F_2$  can be obtained. The force  $F_1$  can be obtained from the free body diagram in Figure 5(b). The sum of the forces acting perpendicularly along the needle must also be zero:

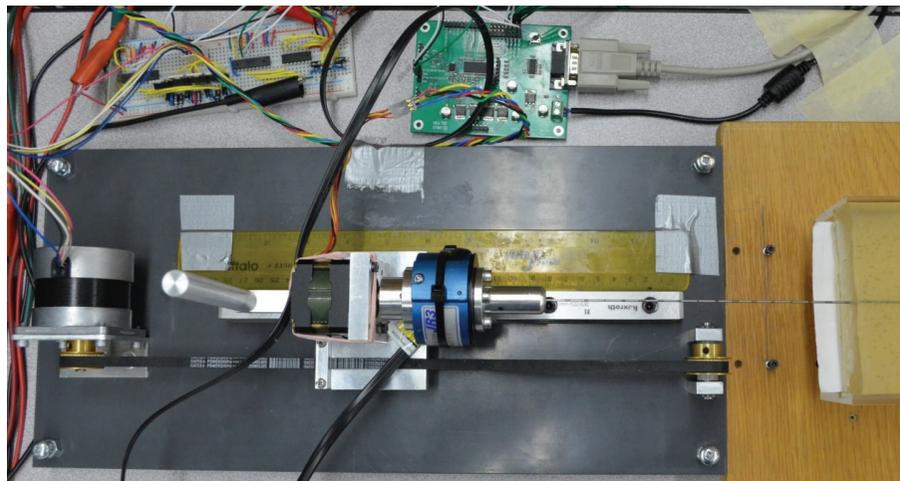
$$\begin{aligned} \sum F &= F_1 - F_R - F_2 = 0 \\ \iff F_1 &= F_R + F_2. \end{aligned} \quad (9)$$

Although here only deflection in the 2D plane is regarded, the model can be easily transferred to 3D, provided that a 4 DOF force/torque sensor is used. Also the deflections can be measured in both directions. If the needle deflects into positive  $y$ -direction, all forces and moments, including forces/torques measured at the needle base, will simply point into the opposite direction.

**2.2. Experimental Setup.** The setup for conducting insertion experiments into soft tissue consists of a robotic system with two degrees of freedom (DOF), which is capable of translational and rotational motion (see Figure 6). The translational motion is along the needle axis and rotational about the needle axis. The restriction to translational rotation along the needle axis is granted by a linear stage, which consists of a ball bearing mounted rail-carriage system. The system provides high precision while maintaining very low friction between rail and carriage (see Figure 6(a)). The carriage, which holds the rotational part, can be connected to a motor, which ensures linear insertion velocities (see Figure 6(b)). To carry out manual insertion experiments, the carriage needs to be entirely decoupled from the motor such that motor inertia does not affect the manual insertion procedure. A belt is used to connect motor and carriage as this provides the



(a) Side view



(b) Top view

FIGURE 6: The setup of the intelligent instrument prototype.

most convenient way to couple or completely decouple motor and carriage by simply clamping the belt to the carriage when necessary. Mounted on the carriage is the motor for needle rotation, which holds a force/torque sensor, and the needle holder, which fixates the hand piece of the needle via two opposing set screws. Also attached to the carriage is a handle to perform manual insertions.

The resulting needle bevel location correction system is a marked improvement over current practice, which typically involves nonoverlapping insertion-only and rotation-only phases; simply making rotations at predetermined insertion depths is an example of the strategies used by implanters. The same is the case in robotic needle insertion research. For instance, in [12], the robot maintains a constant insertion velocity for the needle except when the insertion is paused to rotate the needle by a fixed amount. Or, in [13], high-speed rotation reversal at half the insertion depth was tested. In contrast to the above, we adjust the needle bevel

location during the insertion and believe that such controlled, smooth, and incremental corrections can maximize needle tip targeting accuracy. Our research should not be confused with continuous, drilling-like rotation of a needle during insertion. Given that a bevel-tip needle's trajectory in soft tissue is curved, constantly spinning the needle at a fast rate helps to keep the needle straight but also increases tissue trauma. As opposed to this, we will only adjust the needle's rotational position on a calculated and intelligent basis.

For sensing forces, a JR3 6 DOF sensor of type 50M31A is used. The maximally admissible forces are 100, 100, and 200 N in  $x$ ,  $y$ , and  $z$  direction, and maximum torques are 5 Nm for all three axes. The sensor's ADC has a resolution of 14 bits. The moments are measured about the center of the sensor. Since the relevant moments for deflection measurement occur at the tip of the needle holder (see Figure 6), the measured moments have to be recalculated in our setup. The moment

( $M_H$ ) acting at the tip of the needle holder can be expressed as

$$M_H = M_S - F_S l, \quad (10)$$

where  $M_S$  and  $F_S$  are the moment and force measured by the sensor and  $l$  represents the distance from the center of the force sensor to the tip of the needle holder (52.75 mm).

Both motors for rotational and translational motion are unipolar stepper motors. Stepper motors are used due to their ease of control and availability. The motors are controlled via L297 stepper controller ICs. The clock, enable, and direction signal are provided by a HILINK data acquisition (DAQ) board (see Figure 6(b)). The HILINK board interfaces a PC via RS232 and is controlled in MATLAB/Simulink.

The force sensor is interfaced via a C API. To read the data provided by the force sensor in Simulink, a C S-Function was written. Simulink S-Functions provide a way to run C/C++ code in Simulink models. The S-Function block outputs two 3D vectors for forces and torques. By including the reading of sensor data in Simulink, the setup can entirely be controlled in Simulink. Real time control of the assembly is however only possible in soft real time, as the force sensor's S-Function cannot be used together with MATLAB's Real Time Windows kernel. This limits the maximally admissible sampling rate to roughly 20 Hz, which is a relatively low but sufficient rate for our purposes.

The used phantom tissue for the insertion experiments is liquid plastic, which is made of plastisol and produced by M-F Manufacturing Co. The stiffness of the plastic can be adjusted by the amount of added plastic softener. The thickness of the homogeneous tissue is roughly four centimeters, which provides enough weight to prevent too much shifting along the insertion axis during insertion. A key factor of the phantom tissue is its transparency since the needle needs to be tracked while being inserted in order to measure its deflection. To determine the deflection, images of the needle inside tissue are recorded from above by a Logitech C270 webcam during insertion. Since the needle deflection can only be monitored in the horizontal two-dimensional plane, the deflection also needs to be kept to this plane. This can be achieved by aligning the bevel vertically. This way it can be safely assumed that the needle will deflect in the horizontal plane only. Potentially occurring gravity effects, which could lead to deflection in the vertical plane, can be neglected as the magnitude is below the range of the sensor. Furthermore, the supporting effect of the tissue prevents the needle from being pulled down by gravity.

To track the needle tip during insertion for comparison to estimated deflection, template matching is used. The template is illustrated in Figure 7(a). A MATLAB implementation by *Dirk-Jan Kroon* is used. The used matching method is normalized cross-correlation (NCC). This approach provides a high robustness against brightness variations and specifically bubbles in the tissue (see Figure 7(b), bright spots on the needle). The images have a resolution of  $800 \times 448$  pixels and the camera observes a width of 80 mm at the height of the needle. This results in a resolution of 0.179 mm/pixel. The deflection is measured in respect to the unbent needle in air.

The used needle is a standard 18 G brachytherapy needle with an outer diameter of 1.27 mm and inner diameter of 1 mm. The bevel angle is roughly  $20^\circ$ . The length of the needle is 20 cm, but since a small part of the needle is clamped to the holder, the effective needle length, which can bend, is 19.1 cm.

### 3. Results

In this section we present experimental results for the validation of the proposed deflection model, and for the illustration of the impact that needle turning has on the deflection.

**3.1. Model Verification.** To verify the developed deflection model, insertion experiments were conducted consisting of six trials. Manual insertions with varying velocities and automated insertion with 10 mm/s and 15 mm/s were performed with two homogeneous tissue samples, which varied in stiffness. For this set of experiments, the needle was not turned at any point during the insertion. Forces and moments at the needle base and images were recorded during the insertion. The insertion depth was 120 mm throughout all the trials. Each trial consisted of 6 runs, in each of which a new point of insertion was used and the needle bevel was adjusted to deflect into the right direction as seen from the side of the robot. The recorded forces and moments were then used to estimate the deflection. The measured tip deflection was finally compared to the estimated tip deflection. For the manual insertion it was tried to keep the velocity at a constant level since the insertion speed was not measurable in real time. Before the start of insertion, the needle was inserted approximately 10 mm into the tissue.

Figure 8 shows an unfiltered sample deflection curve of the needle tip (see Figure 8(b)) and forces and moments at the needle base during insertion (see Figure 8(a)). The sensor data was filtered offline by a zero-phase lowpass filter, using MATLAB's *filtfilt*(·) function. Figure 8(b) also shows the deflections estimated by a model proposed by Abolhassani et al. [10] and the estimation of our proposed model. As the plot shows, the estimations of both models are very similar until later in the insertion process. At this point, our model maintains a relatively high precision whereas Abolhassani's model increasingly overestimates the deflection.

In Figures from 9(a) to 9(f), the mean estimations ( $n = 6$ ) at insertion depths of 60 mm and 120 mm are illustrated for all 6 trials. The data shows that the aforementioned overestimation later on during the insertion can be observed throughout all the trials. It furthermore shows that our proposed model also slightly overestimates the deflection at a depth of 120 mm. At a depth of 60 mm, the estimations of both models are very close to the measured deflection throughout most of the trials.

To show where the estimations are in fact significantly different to the measured deflection, a paired  $t$ -test was carried out using MATLAB's function *ttest*(·). The null hypothesis of a paired  $t$ -test is that the mean of the difference of two data sets is *not* significantly different at a significance level of 5%. The test was carried out between measured and estimated samples ( $n = 6$ ) for each model, for each

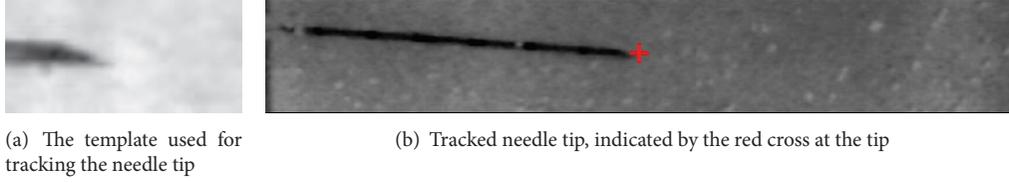


FIGURE 7: Tracking of the needle tip.

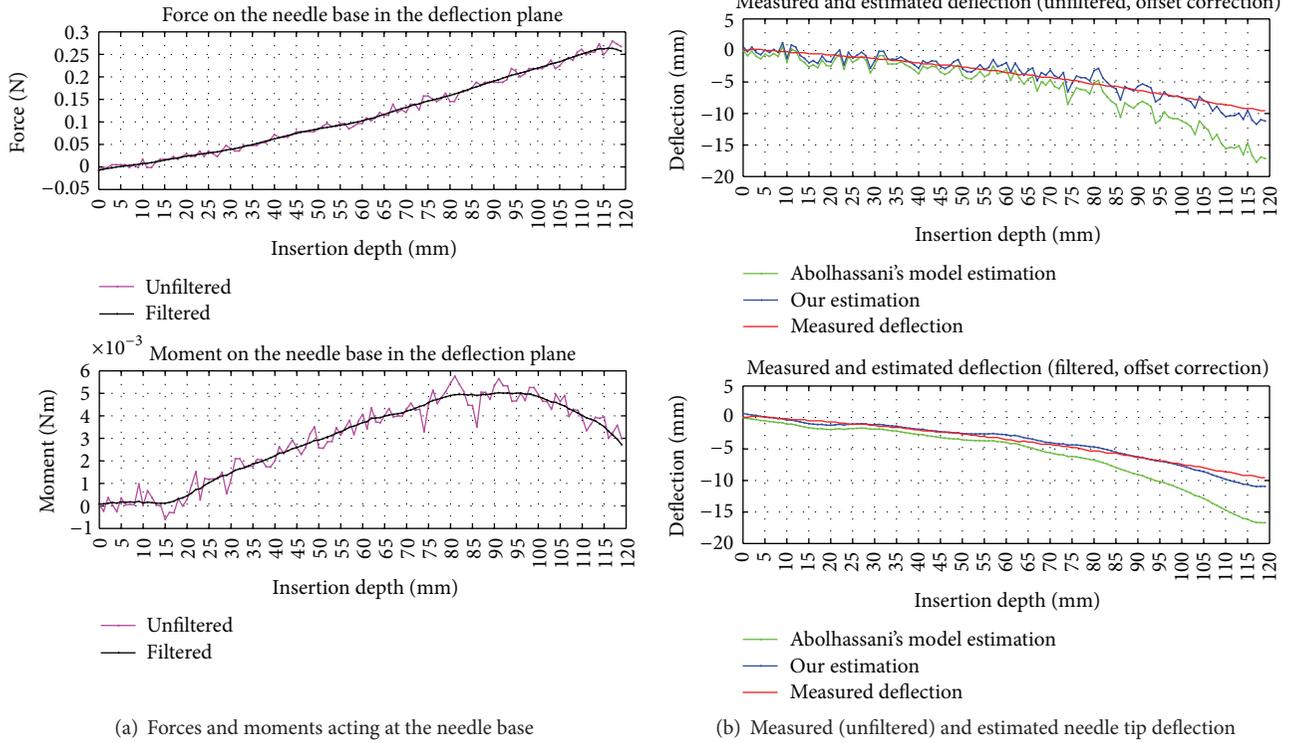


FIGURE 8: Sample plots for forces and moments at the needle base and tip deflection during insertion into tissue 1 with a velocity of 10 mm/s.

trial, and for the two observed insertion depths separately. The results of the  $t$ -test can be found in Table 1. They show that, at the depth of 60 mm, all trials do not reject the null hypothesis, which means that both model estimations are not significantly different at this point. At the depth of 120 mm however, the test shows that both model estimations are significantly different from the measured deflection for most cases. Figures 9(a) to 9(f) and the  $t$ -test confirm the previously made assertion based on Figure 8(b).

**3.2. Correcting the Deflection.** A second type of experiment was performed to study the corrective effect of turning the needle about  $180^\circ$  when a certain estimated deflection threshold is reached. The needle is turned with an angular velocity of 180 deg/s. As the needle is being turned, the insertion is not paused but continues at the same speed. To estimate the threshold, the virtual sensor, which utilizes the developed model, was used. Two trials were executed, also consisting of six runs each. In each trial the threshold was set to a different level, the first being 1 mm and the second 5 mm. The needle also deflected in each run to

TABLE 1: Results of a paired  $t$ -test performed on estimated and measured data.

Model	Tissue 1			Tissue 2		
	10 <sup>a</sup>	15	Man	10	15	Man
Abl <sup>b</sup> 60 <sup>c</sup>	$\bar{r}^d$	$\bar{r}$	$\bar{r}$	$\bar{r}$	$\bar{r}$	$\bar{r}$
Our 60	$\bar{r}$	$\bar{r}$	$\bar{r}$	$\bar{r}$	$\bar{r}$	$\bar{r}$
Abl 120	$r$	$r$	$r$	$r$	$r$	$r$
Our 120	$r$	$\bar{r}$	$r$	$r$	$r$	$r$

<sup>a</sup> Insertion speed in mm/s or “man” for manual.

<sup>b</sup> Model identifier, “Abl” for Abolhassani.

<sup>c</sup> Insertion depth in mm.

<sup>d</sup> “ $\bar{r}$ ” if null hypothesis is *not* rejected.

the right side, or negative  $y$ -direction, until it was turned. One of the tissue samples, which was used for the first experiment (tissue 1), was used for this set of experiments. The runs involving turning the needle were conducted with a velocity of 5 mm/s, and, for the comparison with deflection without turning, data from the model verification

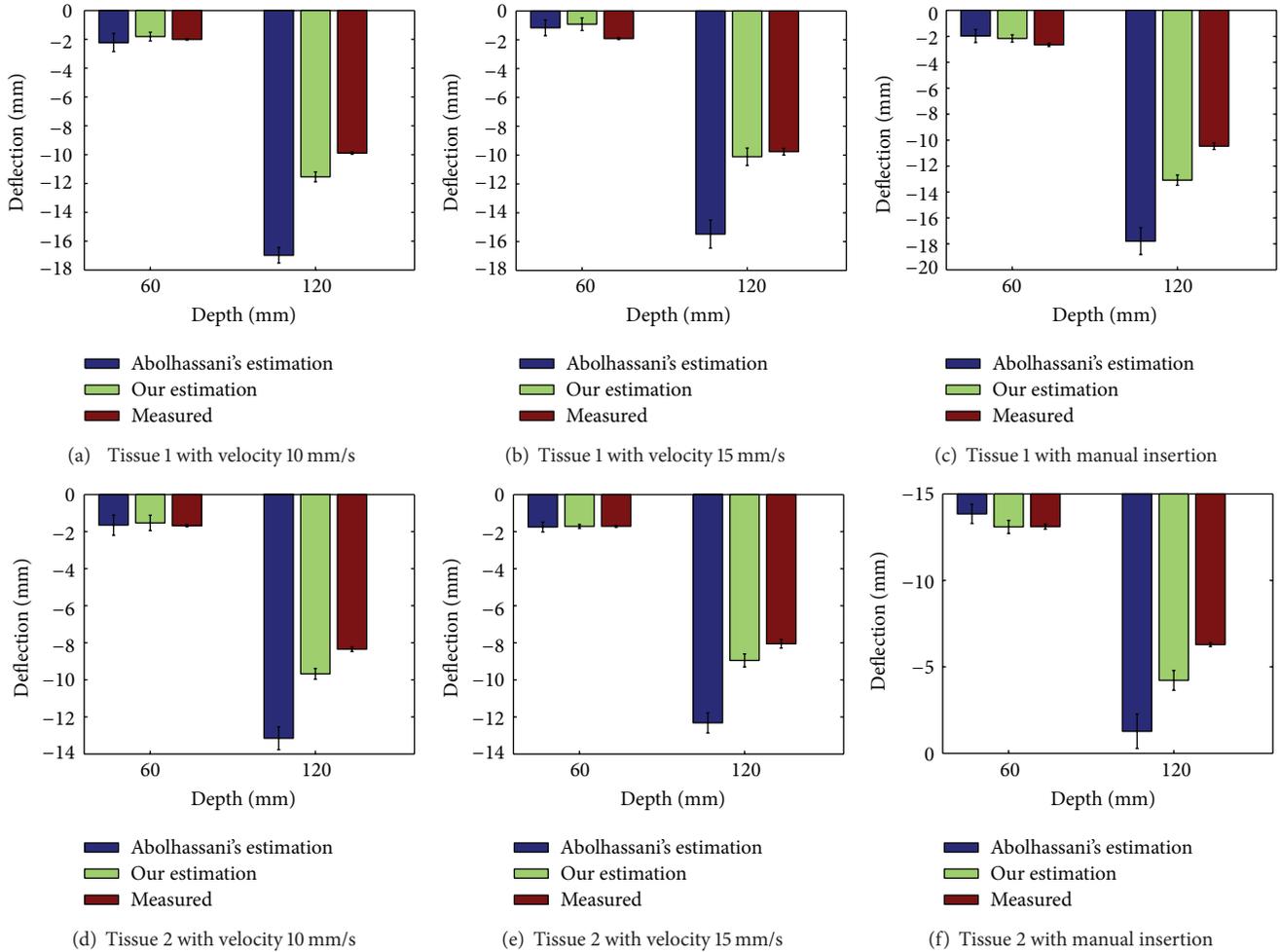


FIGURE 9: Measured and estimated tip deflection with tissue 1 in (a), (b), and (c) and tissue 2 in (d), (e), and (f). The insertion velocities were for (a) and (d) 10 mm/s automatic, for (b) and (e) 15 mm/s automatic, and for (c) and (f) manual with alternating velocities. The error bars show the standard deviation ( $n = 6$ ).

experiment was used. Although the speed differs in between the experiments from 5 to 10 mm/s, judging by Figure 9, the insertion velocity does not influence the amount of deflection. This was also concluded by Webster et al. in 2005 [6].

Real time force readings are used to calculate the deflection with the proposed model in real time. Figure 10(a) shows three insertions: one without rotation, rotated at 5 mm deflection threshold and one rotated at 1 mm deflection threshold. The plot shows that turning the needle at a specific point has a very noticeable impact on the deflection. According to Figure 10(a), the best point for turning the needle is relatively early. The best correction results are achieved when the needle is turned when 1 mm deflection is reached. Figure 10(b) reaffirms this claim. Here the mean ( $n = 6$ ) of each trial (turning at 1 mm, 5 mm and not turning) is shown for an insertion depth of 60 and 120 mm. The error bars, which show the standard deviation, indicate that the results for each turning point are significantly different. None of the error bars overlap.

## 4. Discussion

Several studies have been published, which show the effects of different parameter modifications or relate the force and moment at the needle base to the deflection in soft tissue. Kataoka et al. [3] proposed a first model, which uses a distributed load along the inserted section of the needle to estimate the deflection at the needle tip. The model is based on the Euler-Bernoulli beam theory, which provides a method for calculating the deflection of beams under load. Their results however showed an offset in the estimated tip deflection. Their conclusion was that their model underestimates the deflection outside of tissue, which leads to the offset. Abolhassani et al. [10] proposed a different model, which also relates the forces along the needle to the base forces and moments. Their model is also based on the Euler-Bernoulli beam theory but with different force assumptions along the needle.

Our deflection model, which is based on mechanical properties of the needle introduces a new approach as to

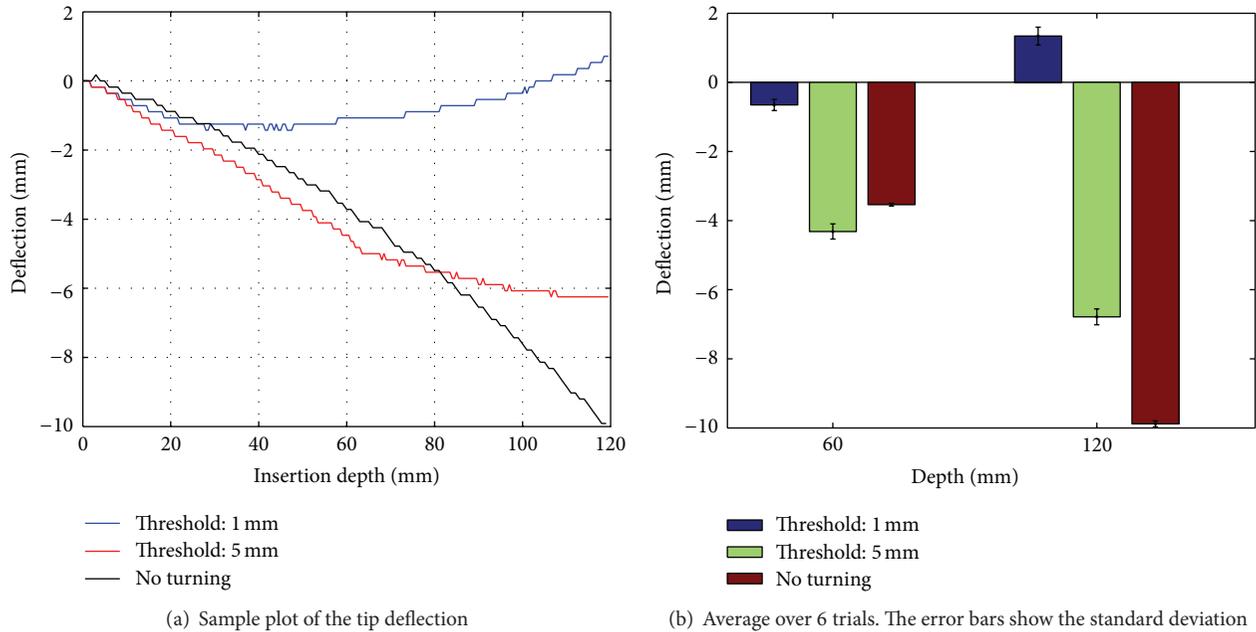


FIGURE 10: The impact of needle turning about  $180^\circ$  at multiple deflection thresholds.

how the forces are composed along the axis of the inserted needle into tissue. This new approach was proven to be more accurate than previously proposed mechanical deflection models.

The distribution of the loads described in Section 2.1, which are applied by the tissue onto the needle, are an assumption, as mentioned previously. Further investigation would be necessary to examine whether the load distribution is triangular or has a rather different shape. If the shape is in fact different, the factors  $a$  and  $b$  in (6) (see also Figure 4(a)) can be adjusted to fit the load distribution. The parameters  $a$  and  $b$  could also be fit to measured data in order to obtain a higher precision for the effective position of  $F_1$  and  $F_2$ . This could further improve the model's performance, especially as the insertion depth increases.

It should be noted that the model can only estimate the deflection while the needle has not been turned. After the needle is turned, the distributed loads along the needle change.

In our experiments, homogeneous phantom tissue samples were used. In our future work, the model's precision will also be tested with tissue which has inhomogeneities.

The results show that a force sensor with a very low range is necessary, as the maximum forces are in the sub-Newton range. The sensor used for these experiments is very close to its lower limit.

## 5. Conclusion

A virtual sensor was introduced in this paper, which is able to precisely sense the needle tip deflection during needle insertion procedures without relying on any other invasive or noninvasive devices than a 4 DOF force sensor, which measures the forces and moments at the needle base. Higher

precisions can be maintained over higher insertion depths with the proposed model than other models before.

It was also shown that adjusting the bevel location during the insertion can drastically increase the tip targeting accuracy with only one rotation during insertion.

This virtual deflection sensor can be applied to a wide range of medical procedures, which involve needles which are prone to deflection.

## Conflict of Interests

The authors of this paper do not have any affiliations to or relationships with the companies NDI, M-F Manufacturing Co., JR3 Inc., Zeltom, Logitech, and Mathworks.

## References

- [1] R. S. Sloboda, N. Usmani, J. Pedersen, A. Murtha, N. Pervez, and D. Yee, "Time course of prostatic edema post permanent seed implant determined by magnetic resonance imaging," *Brachytherapy*, vol. 9, no. 4, pp. 354–361, 2010.
- [2] V. Lagerburg, M. A. Moerland, J. J. W. Lagendijk, and J. J. Battermann, "Measurement of prostate rotation during insertion of needles for brachytherapy," *Radiotherapy and Oncology*, vol. 77, no. 3, pp. 318–323, 2005.
- [3] H. Kataoka, T. Washio, M. Audette, and K. Mizuhara, "A model for relations between needle deflection, force, and thickness on needle penetration," in *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI '01)*, pp. 966–974, Springer, Heidelberg, Germany, 2001.
- [4] R. Alterovitz, A. Lim, K. Goldberg, G. S. Chirikjian, and A. M. Okamura, "Steering flexible needles under Markov motion uncertainty," in *Proceedings of the IEEE IRS/RSJ International*

- Conference on Intelligent Robots and Systems (IROS '05)*, pp. 120–125, August 2005.
- [5] A. M. Okamura, C. Simone, and M. D. O’Leary, “Force modeling for needle insertion into soft tissue,” *IEEE Transactions on Bio-Medical Engineering*, vol. 51, no. 10, pp. 1707–1716, 2004.
  - [6] R. J. Webster, J. Memisevic, and A. M. Okamura, “Design considerations for robotic needle steering,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '05)*, pp. 3588–3594, April 2005.
  - [7] R. J. Webster III, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, “Nonholonomic modeling of needle steering,” *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 509–525, 2006.
  - [8] N. Abolhassani, R. Patel, and M. Moallem, “Trajectory generation for robotic needle insertion in soft tissue,” in *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 4, 2004.
  - [9] R. Hibbeler, *Engineering Mechanics: Statics*, Engineering Mechanics, Prentice Hall, 6th edition, 2010.
  - [10] N. Abolhassani, R. Patel, and F. Ayazi, “Needle control along desired tracks in robotic prostate brachytherapy,” in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (ISIC '07)*, pp. 3361–3366, October 2007.
  - [11] J. M. Gere and B. J. Goodno, *Mechanics of Materials*, CL Engineering, 2012.
  - [12] N. Abolhassani, R. V. Patel, and F. Ayazi, “Minimization of needle deflection in robot-assisted percutaneous therapy,” *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 3, no. 2, pp. 140–148, 2007.
  - [13] G. Wan, Z. Wei, L. Gardi, D. B. Downey, and A. Fenster, “Brachytherapy needle deflection evaluation and correction,” *Medical Physics*, vol. 32, no. 4, pp. 902–909, 2005.

## Research Article

# Grey-Level Cooccurrence Matrix Performance Evaluation for Heading Angle Estimation of Moveable Vision System in Static Environment

Zairulazha Zainal,<sup>1,2</sup> Rizauddin Ramli,<sup>1</sup> and Mohd Marzuki Mustafa<sup>1</sup>

<sup>1</sup> Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

<sup>2</sup> Faculty of Mechanical Engineering, Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia

Correspondence should be addressed to Zairulazha Zainal; [zazain@vlsi.eng.ukm.my](mailto:zazain@vlsi.eng.ukm.my)

Received 14 March 2013; Accepted 7 May 2013

Academic Editor: Aiguo Song

Copyright © 2013 Zairulazha Zainal et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A method of extracting information in estimating heading angle of vision system is presented. Integration of grey-level cooccurrence matrix (GLCM) in an area of interest selection is carried out to choose a suitable region that is feasible for optical flow generation. The selected area is employed for optical flow generation by using Horn-Schunck method. From the generated optical flow, heading angle is estimated and enhanced via moving median filter (MMF). In order to ascertain the effectiveness of GLCM, we compared the result with a different estimation method of optical flow which is generated directly from untouched greyscale images. The performance of GLCM is compared to the true heading, and the error is evaluated through mean absolute deviation (MAE). The result ensured that GLCM can improve the estimation result of the heading angle of vision system significantly.

## 1. Introduction

Sensors are most the important components in measurement. With the data gathered from sensor, data analysis as well as control strategy implementation can be conducted. Due to the importance of the sensor, there are various works conducted in this field to cater issues that arise in this field, for instance, new design of force sensor [1, 2] and tactile display [3].

Particularly in the field of robotics, sensors provide information about a variable been measured in order to control the robot, which is most critical part. This information is processed in order to decide the way the robot should take an action with good performance. There are various kinds of sensors available for this purpose, for example, sonar sensor, position sensor, infrared sensor, and camera. Employment of a camera as a sensor, also known as robot vision system, is an interesting idea due to its capabilities to obtain useful information from surroundings through image acquisition. Depending on the application, extraction of information gathered from robot vision system can be

executed in various ways. The extraction process commonly utilises image processing and analysis.

A research in the field of vision system is ranging from hardware development and testing, algorithm development specifically on image processing and analysis, and strategy proposal regarding the way to use vision system for a specific task and its implementation. Conducted previous works include manipulation of image acquisition capabilities of the optical mouse to be used as a sensor [4–6], utilization of panoramic image in visual navigation [7], log-polar imaging application in robotic vision [8], application of sparse visual data without odometry reading [9], obstacle avoidance strategy [10], visual mapping [11], and strategy for robust real-time vision [12].

In this paper, a new approach has been proposed in extracting information regarding the heading angle of vision system using optical flow, with the belief that this work will be beneficial especially in robot navigation. Since it is well understood that some flow vectors generated using this method also contained erroneous vectors which are not representing actual motion, it is necessary to improve the

interpretation by integrating a new approach. Here, we proposed the use of grey-level cooccurrence matrix (GLCM) [13] together with optical flow computation as an improvement technique in heading angle estimation. By dividing the image into several areas of interest (AOI), the selection of the most suitable area will be handled by GLCM before an optical flow calculation is made. Additionally, from generated an optical flow vector, heading angle estimation is implemented, and this estimation is further smothered using moving median filter (MMF) in order to eliminate the outlier existed in estimation. The estimation performance is carried out using mean absolute deviation (MAE), and the results from the proposed technique are compared with the estimation result using unmodified optical flow vector field.

## 2. Grey-Level Cooccurrence Matrix

Grey-level cooccurrence matrix is one technique in texture feature extraction [14]. In general, texture in image shows a repeating pattern of local variation in intensity. It provides information in the spatial arrangement of colours or intensities in an image, characterized by the spatial distribution of intensity levels in a neighbourhood. GLCM is implemented based on statistical approach, where the texture is manipulated as a quantitative measure of the arrangement of intensities in a region. Various works have been done which take a benefit of GLCM, such as in [15–18].

Specifically, GLCM is defined as tabulation of occurrence frequency with regarding to a different combination of pixel intensity values in an image. In this method, consideration has been given to the relation between two pixels at a time which is called a reference and neighbour pixel. This relationship is displayed in the form of cooccurrence matrix. Turning it into an equation, it can be described as

$$P_d [i, j] = n_{ij}. \quad (1)$$

Here,  $n_{ij}$  is denoted as the number of occurrence of the pixel values  $(i, j)$  posses at length  $d$  in the image. The occurrence matrix  $P_d$  has dimension  $n \times n$ , where  $n$  is denoted as the number of grey levels in the image. The grey levels in the image are specified under quantization. The elements of  $P_d$  are then normalized using the following equation:

$$N [i, j] = \frac{P [i, j]}{\sum_{i=1}^n \sum_{j=1}^n P [i, j]}. \quad (2)$$

Through normalization, elements in matrix  $P_d$  have a value which is in range between zero and one, which allows to be manipulated as a function of probability.

Through GLCM, texture properties can be captured. Using cooccurrence matrix, feature vector can be extracted in several ways. In this paper, contrast has been selected in extracting feature vector. Contrast is a measure of the local variations present in an image, which are represented as

$$C (k, n) = \sum_{i=1}^n \sum_{j=1}^n (i - j)^2 N [i, j]. \quad (3)$$

## 3. Optical Flow

Optical flow technique is commonly used in motion detection and estimation. Optical flow equation is derived based on the assumption that the brightness is preserved unchanged over an entire image sequence and is represented as

$$I_x u + I_y v + I_t = 0. \quad (4)$$

Here  $I_x, I_y, I_t$  are denoted as spatial and temporal gradient which can be estimated from image sequence. Meanwhile  $u, v$  are denoted as optical flow vector and they are unknowns. Since optical flow equation can be acquired from only a single pixel, hence, these unknowns cannot be solved directly. Regarding this matter, several methods have been proposed in helping to find a solution of these unknowns. In this paper, the method that has been selected is Horn-Schunck method [19].

In Horn-Schunck method, in order to solve these unknowns, optimization constraint is introduced with the assumption that

(A1) motion stays constant within pixel neighbourhood;

(A2) optical flow differs smoothly.

The optimization constraint is defined as

$$E_t = (I_x u + I_y v + I_t)^2 + \zeta (u_x^2 + u_y^2 + v_x^2 + v_y^2), \quad (5)$$

$$\zeta > 0.$$

Here,  $\zeta$  is a weighting factor, and  $u_x, u_y, v_x, v_y$  are partial derivatives of the optical components  $u$  and  $v$ . Moreover, computation of  $I_x, I_y, I_t$  is implemented using the following equations:

$$I_x \approx \frac{1}{4} \{ I_{(n_1+1, n_2, k)} - I_{(n_1, n_2, k)} + I_{(n_1+1, n_2+1, k)} - I_{(n_1, n_2+1, k)} + I_{(n_1+1, n_2, k+1)} - I_{(n_1, n_2, k+1)} + I_{(n_1+1, n_2+1, k+1)} - I_{(n_1, n_2+1, k+1)} \},$$

$$I_y \approx \frac{1}{4} \{ I_{(n_1, n_2+1, k)} - I_{(n_1, n_2, k)} + I_{(n_1+1, n_2+1, k)} - I_{(n_1+1, n_2, k)} + I_{(n_1, n_2+1, k+1)} - I_{(n_1, n_2, k+1)} + I_{(n_1+1, n_2+1, k+1)} - I_{(n_1+1, n_2, k+1)} \},$$

$$I_t \approx \frac{1}{4} \{ I_{(n_1, n_2, k+1)} - I_{(n_1, n_2, k)} + I_{(n_1+1, n_2, k+1)} - I_{(n_1+1, n_2, k)} + I_{(n_1, n_2+1, k+1)} - I_{(n_1, n_2+1, k)} + I_{(n_1+1, n_2+1, k+1)} - I_{(n_1+1, n_2+1, k)} \}.$$

In terms of partial derivatives of the optical components  $u$  and  $v$ , it can be computed from the 4-adjacent position to the position  $(i, j)$ . Therefore,

$$\begin{aligned} u_x^2 + u_y^2 + v_x^2 + v_y^2 &= \frac{1}{4} \left[ \left\{ u_{(i,j)} - u_{(i-1,j)} \right\}^2 + \left\{ u_{(i+1,j)} - u_{(i,j)} \right\}^2 \right. \\ &\quad + \left\{ u_{(i,j+1)} - u_{(i,j)} \right\}^2 + \left\{ u_{(i,j)} - u_{(i,j-1)} \right\}^2 \\ &\quad + \left\{ v_{(i,j)} - v_{(i-1,j)} \right\}^2 + \left\{ v_{(i+1,j)} - v_{(i,j)} \right\}^2 \\ &\quad \left. + \left\{ v_{(i,j+1)} - v_{(i,j)} \right\}^2 + \left\{ v_{(i,j)} - v_{(i,j-1)} \right\}^2 \right]. \end{aligned} \quad (7)$$

In addition, through differentiating and minimizing (5) over  $u$  and  $v$  at position  $(i, j)$ , the equation can be rewritten as

$$\begin{aligned} \frac{\partial E_t}{\partial u_{(i,j)}} &= 2 \left( u_{(i,j)} \cdot I_x(i, j) + v_{(i,j)} \cdot I_y(i, j) + I_t(i, j) \right) \\ &\quad \times I_x(i, j) + 2\zeta \left( u_{(i,j)} - \bar{u}_{(i,j)} \right) = 0, \end{aligned} \quad (8)$$

$$\begin{aligned} \frac{\partial E_t}{\partial v_{(i,j)}} &= 2 \left( u_{(i,j)} \cdot I_x(i, j) + v_{(i,j)} \cdot I_y(i, j) + I_t(i, j) \right) \\ &\quad \times I_y(i, j) + 2\zeta \left( v_{(i,j)} - \bar{v}_{(i,j)} \right) = 0. \end{aligned}$$

Here,

$$\begin{aligned} \bar{u}_{(i,j)} &= \frac{1}{4} \left( u_{(i-1,j)} + u_{(i+1,j)} + u_{(i,j+1)} + u_{(i,j-1)} \right), \\ \bar{v}_{(i,j)} &= \frac{1}{4} \left( v_{(i-1,j)} + v_{(i+1,j)} + v_{(i,j+1)} + v_{(i,j-1)} \right). \end{aligned} \quad (9)$$

From (8), the unknowns are  $u_{(i,j)}$  and  $v_{(i,j)}$ . The relationship between  $\bar{u}_{(i,j)}$ ,  $\bar{v}_{(i,j)}$ , and  $u_{(i,j)}$ ,  $v_{(i,j)}$  can be observed. Therefore, iterative scheme can be constructed based on this relationship, as shown in the following equation:

$$\begin{aligned} u^{n+1} &= \bar{u}^n - I_x \cdot \frac{I_x \bar{u}^n + I_y \bar{v}^n + I_t}{\zeta^2 + (I_x)^2 + (I_y)^2}, \\ v^{n+1} &= \bar{v}^n - I_y \cdot \frac{I_x \bar{u}^n + I_y \bar{v}^n + I_t}{\zeta^2 + (I_x)^2 + (I_y)^2}. \end{aligned} \quad (10)$$

In (10),  $n$  is described as iteration step, and  $\bar{u}$ ,  $\bar{v}$  are described as the neighbourhood mean of the flow vector  $u$ ,  $v$ , respectively.

#### 4. Heading Angle Estimation Strategy

In this study, a static environment is selected as an experimental scenario. A camera is shifted over horizontal flat surface from left to right, as shown in Figure 1.

Using this camera, a video is captured, and the image sequence is extracted. Extracted images have a resolution of  $320 \times 176$  pixels. At initial stage, all images are converted



FIGURE 1: Camera movement from left to right in static environment.

to greyscale format and then resized to  $99 \times 99$  pixels. Each converted image is divided into nine AOI. Since GLCM is employed, the intensity of the image needs to be scaled to several grey levels. In this study, the image has been scaled to eight grey levels. In each AOI, the GLCM is generated, and a feature vector is extracted using contrast criterion. By using computed feature vector, a suitable AOI is selected.

In the next stage, optical flow field vector is generated using Horn-Schunck method. Vector operation is employed where each flow vector is decoupled to horizontal and vertical components. Summation of each decoupled vector regarding horizontal and vertical axes is implemented, and angle estimation is computed. MMF is applied to acquire smoothness of the estimated angle.

In order to assess the estimation performance, the evaluation criterion used is mean absolute error (MAE). Selection of this criterion is due to the fact that MAE is more robust to outliers, compared to other popular methods such as root mean square error (RMSE) [20]. For that purpose, estimation gathered from the proposed method is compared with the actual value, which is  $180^\circ$ . MAE is defined as follows:

$$\text{MAE} = \frac{1}{M} \sum_{i=1}^M |e_i|, \quad (11)$$

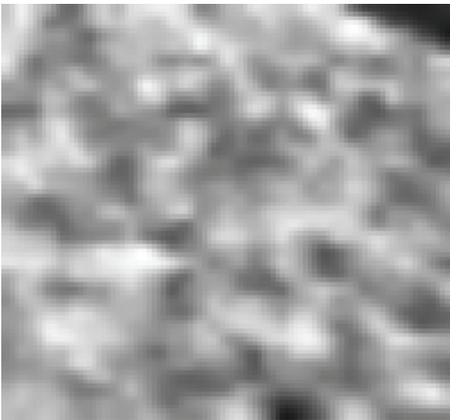
where

$$e_i = f_i - y_i. \quad (12)$$

Here,  $f_i$  is the estimation value and  $y_i$  is the true value.

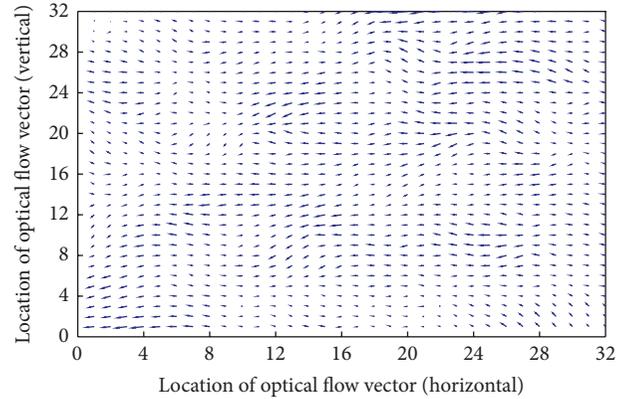
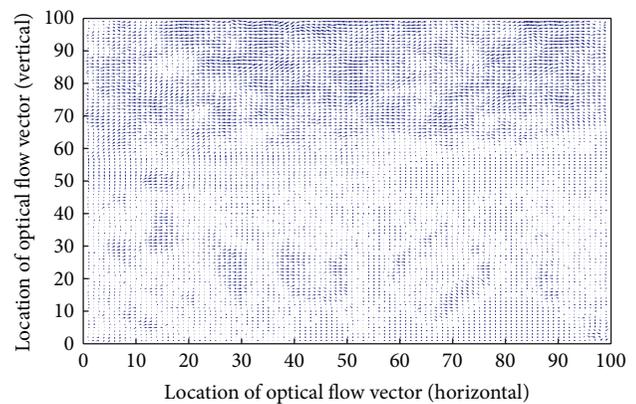
#### 5. Results and Discussion

In this experiment, ten images were captured by moving the camera in each second. One hundred images have been

FIGURE 2: Image at  $t = 5.6$  s.FIGURE 3: Selected AOI of the image at  $t = 5.6$  s.

captured. The average processing time, starting from image preprocessing until angle estimation, is 0.02 sec. This is the average value based on the computation time to process one hundred images. This processing time is much less than the image sampling time of 0.1 sec. By using GLCM and feature vector extraction based on contrast, the AOI was selected for each image. The example of selected AOI is shown in Figures 2 and 3.

In the AOI selection, the values of feature vectors were calculated, and median value was selected. The reason why median value was decided to be used is because, after operation of GLCM and contrast, diagonal value of cooccurrence matrix is equal to zero, which means that similar grey-level values, occurred next to one another, were excluded. It is also understood that higher contrast value is caused by the effect of large contrast weights. This shows that large deviation on grey-level values occurred next to one another. In this test, the image contrast changed along with the camera movement. This is due to the changes in the image texture, which was affected by the changes of environment structure. Therefore, the AOI which shows the median value among the values generated by GLCM and contrast will be selected. However, the decision is made under the assumption that the

FIGURE 4: Optical flow vector field using selected AOI at  $t = 5.6$  s.FIGURE 5: Optical flow vector field using an image without AOI at  $t = 5.6$  s.

contrast in the selected AOI does not have significant contrast difference over the time.

As a result of using the selected AOI, computed optical flow field is shown in Figure 4.

In addition, for comparison purpose, optical flow field created using an image without AOI is shown in Figure 5.

In Figure 5, there exist vectors whose directions are scattered. It is also observed that there are vectors whose its magnitude is too small. These types of vectors are called mislead vectors and irrelevant vectors, respectively. Mislead vectors may cause large errors in estimation because it is not related to an actual motion. The reason why these vectors exist is because of an assumption of brightness constancy, where, in a real situation, this is not easily maintained, especially in outdoor environment. On the other hand, irrelevant vectors show that there is almost small or no motion detected at vectors location. This is due to image texture at that location, which is remaining constant. For that reason, through the proposed method, improvement in optical flow generation is discovered.

Using these results, heading angle estimation is figured out. The estimation results are given in Figures 6 and 7, accordingly.

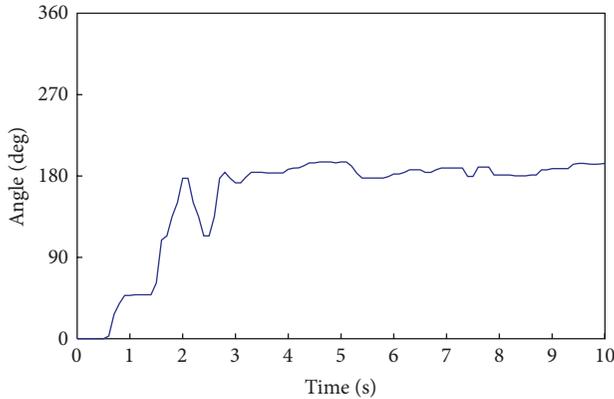


FIGURE 6: Angle estimation using the proposed method.

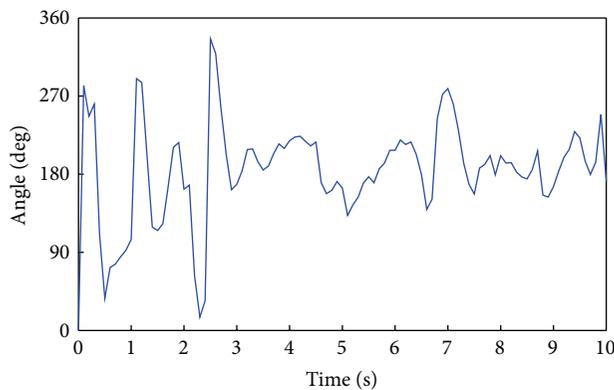


FIGURE 7: Angle estimation without using the proposed method.

From Figures 6 and 7, it can be noticed that heading angle estimation shows an improvement when proposed GLCM is applied. Without GLCM, as shown in Figure 7, the estimation fluctuated over time, where extremely large spikes occurred that caused to poor estimation. Hence, through GLCM method, significant oscillation reduction is observed which makes the estimation result improved.

Related to estimation performance, Table 1 shows the results of evaluation conducted using criteria that have been mentioned in Section 4.

From Table 1, the evaluation results of the heading angle estimation which incorporates GLCM show better performance. MAE result without GLCM shows large deviation existence between estimation and actual value. It is because calculated absolute error is contributed from numbers of large oscillations as well as mediocre but rapid oscillations, as shown in Figure 7. On the other hand, the MAE value for proposed GLCM is smaller. This small MAE value is due to the estimation result in Figure 6 which shows large deviations between estimation and actual value that occurred at the beginning, and the deviations become smaller as the estimation becomes stable and reaches actual value. Therefore, most errors that contributed to MAE are from large variations between estimations and actual values at the beginning of the experiment.

TABLE 1: Performance evaluation using mean absolute error (MAE).

Method	Mean absolute error (degree)
Without AOI	40.9
With AOI	32.5

In addition, estimation results without GLCM in Figure 7 show that the oscillation which occurred frequently caused difficulties to identify the heading angle of the vision system. This problem can be solved when our proposed GLCM method is applied.

## 6. Conclusion

In this paper, the effectiveness of the proposed method in estimating heading angle of vision system is presented. Each captured image has been divided into nine AOI, and GLCM contrast has been used to extract feature vector in order to identify the most suitable region that can be used for estimation purpose. Optical flow calculated using Horn-Schunck method has been used as a basis in extracting required information. Using generated optical flow, estimation on angle has been implemented and smothered using MMF. The result shows that an improvement in estimation is observed when compared to an angle estimation based on untouched optical flow. Through the proposed method, invaluable information regarding the system's current status can be obtained.

## Acknowledgments

The authors acknowledge the Ministry of Higher Education Malaysia, Universiti Kebangsaan Malaysia, and Universiti Teknikal Malaysia Melaka for their sponsorship in this study.

## References

- [1] A. Song, J. Wu, G. Qin, and W. Huang, "A novel self-decoupled four degree-of-freedom wrist force/torque sensor," *Measurement*, vol. 40, no. 9-10, pp. 883-891, 2007.
- [2] G. Song, H. Yuan, Y. Tang, Q. Song, and Y. Ge, "A novel three-axis force sensor for advanced training of shot-put athletes," *Sensors and Actuators A*, vol. 128, no. 1, pp. 60-65, 2006.
- [3] J. Wu, Z. Song, W. Wu, A. Song, and D. Constantinescu, "A vibro-tactile system for image contour display," in *Proceedings of the IEEE International Symposium on Virtual Reality Innovations (ISVRI '11)*, pp. 145-150, March 2011.
- [4] J. Palacin, I. Valgañon, and R. Pernia, "The optical mouse for indoor mobile robot odometry measurement," *Sensors and Actuators A*, vol. 126, no. 1, pp. 141-147, 2006.
- [5] M. Tresanchez, T. Pallejà, M. Teixidó, and J. Palacín, "The optical mouse sensor as an incremental rotary encoder," *Sensors and Actuators A*, vol. 155, no. 1, pp. 73-81, 2009.
- [6] M. Tresanchez, T. Pallejà, M. Teixidó, and J. Palacín, "Using the image acquisition capabilities of the optical mouse sensor to build an absolute rotary encoder," *Sensors and Actuators A*, vol. 157, no. 1, pp. 161-167, 2010.

- [7] F. Labrosse, "Short and long-range visual navigation using warped panoramic images," *Robotics and Autonomous Systems*, vol. 55, no. 9, pp. 675–684, 2007.
- [8] V. Javier Traver and A. Bernardino, "A review of log-polar imaging for visual perception in robotics," *Robotics and Autonomous Systems*, vol. 58, no. 4, pp. 378–398, 2010.
- [9] M. Kronfeld, C. Weiss, and A. Zell, "Swarm-supported outdoor localization with sparse visual data," *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 166–173, 2010.
- [10] K. Souhila and A. Karim, "Optical flow based robot obstacle avoidance," *International Journal of Advanced Robotic Systems*, vol. 4, no. 1, pp. 13–16, 2007.
- [11] L. Wang, H. Li, and R. Hartley, "Video local pattern based image matching for visual mapping," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, vol. 2, pp. 67–70, 2006.
- [12] G. Medioni, A. R. J. François, M. Siddiqui, K. Kim, and H. Yoon, "Robust real-time vision for a personal service robot," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 196–203, 2007.
- [13] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.
- [14] V. Bino Sebastian, A. Unnikrishnan, and K. Balakrishnan, "Grey level co-occurrence matrices: generalisation and some new features," *International Journal of Computer Science, Engineering and Information Technology*, vol. 2, no. 2, pp. 151–157, 2012.
- [15] K. Padmavasavi, N. U. Kumar, E. V. K. Rao, and M. Madhavalatha, "Performance evaluation of adaptive statistical thresholding based edge detection using GLCM in wavelet domain under noisy conditions," *ICGST International Journal on Graphics, Vision and Image Processing (GVIP)*, vol. 10, no. 3, pp. 35–44, 2010.
- [16] M. M. Mokji and S. A. R. S. A. Bakar, "Adaptive thresholding based on co-occurrence matrix edge information," *Journal of Computers*, vol. 2, no. 8, pp. 44–52, 2007.
- [17] K. H. Ghazali, M. M. Mustafa, A. Hussain, and F. Engineering, "Machine vision system for automatic weeding strategy using image processing technique," *American-Eurasian Journal of Agricultural & Environmental Sciences*, vol. 3, no. 3, pp. 451–458, 2008.
- [18] C. Lane, R. L. Burguete, and A. Shterenlikht, "An objective criterion for the selection of an optimum DIC pattern and subset size," in *Proceedings of the 11th International Congress and Exhibition on Experimental and Applied Mechanics*, pp. 900–908, June 2008.
- [19] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–203, 1981.
- [20] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.