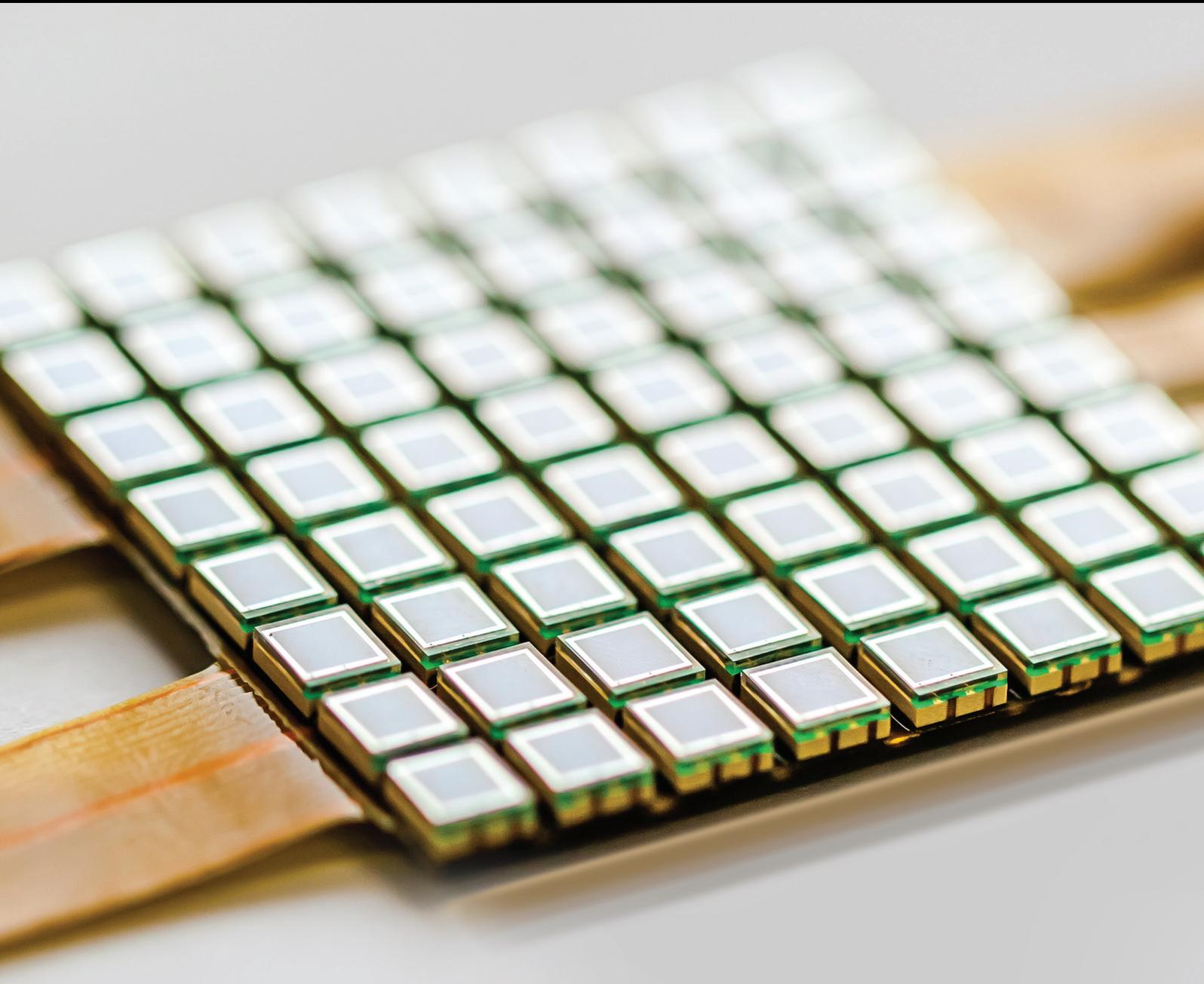


# Sensors Technologies and Methods for Perception Systems in Intelligent Vehicles

Guest Editors: Yassine Ruichek, Fadi Dornaika, and Maan El Badaoui El Najjar





---

# **Sensors Technologies and Methods for Perception Systems in Intelligent Vehicles**

# **Sensors Technologies and Methods for Perception Systems in Intelligent Vehicles**

Guest Editors: Yassine Ruichek, Fadi Dornaika,  
and Maan El Badaoui El Najjar



---

Copyright © 2016 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

Harith Ahmad, Malaysia  
Francisco J. Arregui, Spain  
Francesco Baldini, Italy  
Fernando Benito-Lopez, Ireland  
Romeo Bernini, Italy  
Shekhar Bhansali, USA  
Wojtek J. Bock, Canada  
Hubert Brändle, Switzerland  
Davide Brunelli, Italy  
Paolo Bruschi, Italy  
Belén Calvo, Spain  
Stefania Campopiano, Italy  
Domenico Caputo, Italy  
Sara Casciati, Italy  
Gabriele Cazzulani, Italy  
Chi Chiu Chan, Singapore  
Nick Chaniotakis, Greece  
Nicola Cioffi, Italy  
Elisabetta Comini, Italy  
Marco Consales, Italy  
Jesus Corres, Spain  
Andrea Cusano, Italy  
Antonello Cutolo, Italy  
Dzung V. Dao, Japan  
Manel del Valle, Spain  
Ignacio Del Villar, Spain  
Francesco Dell'Olio, Italy  
Utkan Demirci, USA  
Nicola Donato, Italy  
Junhang Dong, USA  
Abdelhamid Errachid, France  
Stephane Evoy, Canada  
Vittorio Ferrari, Italy  
Luca Francioso, Italy  
Laurent Francis, Belgium  
Mohammad Reza Ganjali, Iran  
Wei Gao, Japan  
Michele Giordano, Italy  
Banshi D. Gupta, India  
Clemens Heitzinger, Austria  
María del Carmen Horrillo, Spain  
Wieslaw Jakubik, Poland  
Hai-Feng Ji, USA  
Kouros Kalantar-Zadeh, Australia  
Sher Bahadar Khan, KSA  
Sang Sub Kim, Republic of Korea  
Challa Kumar, USA  
Laura M. Lechuga, Spain  
Chengkuo Lee, Singapore  
Chenzhong Li, USA  
Eduard Llobet, Spain  
Yu-Lung Lo, Taiwan  
Oleg Lupan, Moldova  
Frederick Maily, France  
Eugenio Martinelli, Italy  
Jose R. Martinez-de Dios, Spain  
Yasuko Y. Maruo, Japan  
Mike McShane, USA  
Igor L. Medintz, USA  
Fanli Meng, China  
Aldo Minardo, Italy  
Joan Ramon Morante, Spain  
Lucia Mosiello, Italy  
Masayuki Nakamura, Japan  
Calogero M. Oddo, Italy  
Marimuthu Palaniswami, Australia  
Alberto J. Palma, Spain  
Lucio Pancheri, Italy  
Alain Pauly, France  
Giorgio Pennazza, Italy  
Michele Penza, Italy  
Andrea Ponzoni, Italy  
Biswajeet Pradhan, Malaysia  
Ioannis Raptis, Greece  
Armando Ricciardi, Italy  
Christos Riziotis, Greece  
Maria Luz Rodríguez-Méndez, Spain  
Albert Romano-Rodriguez, Spain  
Carlos Ruiz, Spain  
Josep Samitier, Spain  
Giorgio Sberveglieri, Italy  
Luca Schenato, Italy  
Michael J. Schöning, Germany  
Andreas Schütze, Germany  
Woosuck Shin, Japan  
Pietro Siciliano, Italy  
Vincenzo Spagnolo, Italy  
Vincenzo Stornelli, Italy  
Weilian Su, USA  
Tong Sun, United Kingdom  
Raymond Swartz, USA  
Hidekuni Takao, Japan  
Isao Takayanagi, Japan  
Guiyun Tian, United Kingdom  
Suna Timur, Turkey  
Hana Vaisocherova, Czech Republic  
Qihao Weng, USA  
Matthew J. Whelan, USA  
Stanley E. Woodard, USA  
Hai Xiao, USA

# Contents

---

## **Sensors Technologies and Methods for Perception Systems in Intelligent Vehicles**

Yassine Ruichek, Fadi Dornaika, and Maan El Badaoui El Najjar  
Volume 2016, Article ID 7241243, 1 page

## **FastSLAM Using Compressed Occupancy Grids**

Christopher Cain and Alexander Leonessa  
Volume 2016, Article ID 3891865, 23 pages

## **A New Technique for Integrating MEMS-Based Low-Cost IMU and GPS in Vehicular Navigation**

Neda Navidi, René Jr. Landry, Jianhua Cheng, and Denis Gingras  
Volume 2016, Article ID 5365983, 16 pages

## **Bayesian Train Localization with Particle Filter, Loosely Coupled GNSS, IMU, and a Track Map**

Oliver Heirich  
Volume 2016, Article ID 2672640, 15 pages

## **A Novel Improved Probability-Guided RANSAC Algorithm for Robot 3D Map Building**

Songmin Jia, Ke Wang, Xiuzhi Li, and Tao Xu  
Volume 2016, Article ID 3243842, 18 pages

## **An Analytical Measuring Rectification Algorithm of Monocular Systems in Dynamic Environment**

Deshi Li and Xiaoliang Wang  
Volume 2016, Article ID 4132721, 9 pages

## **Using Omnidirectional Vision to Create a Model of the Environment: A Comparative Evaluation of Global-Appearance Descriptors**

L. Payá, O. Reinoso, Y. Berenguer, and D. Úbeda  
Volume 2016, Article ID 1209507, 21 pages

## **Optical Flow Sensor/INS/Magnetometer Integrated Navigation System for MAV in GPS-Denied Environment**

Chong Shen, Zesen Bai, Huiliang Cao, Ke Xu, Chenguang Wang, Huaiyu Zhang, Ding Wang, Jun Tang, and Jun Liu  
Volume 2016, Article ID 6105803, 10 pages

## **Intraframe Scene Capturing and Speed Measurement Based on Superimposed Image: New Sensor Concept for Vehicle Speed Measurement**

Mate Nemeth and Akos Zarandy  
Volume 2016, Article ID 8696702, 10 pages

## **Night-Time Vehicle Sensing in Far Infrared Image with Deep Learning**

Hai Wang, Yingfeng Cai, Xiaobo Chen, and Long Chen  
Volume 2016, Article ID 3403451, 8 pages

## *Editorial*

# **Sensors Technologies and Methods for Perception Systems in Intelligent Vehicles**

**Yassine Ruichek,<sup>1</sup> Fadi Dornaika,<sup>2,3</sup> and Maan El Badaoui El Najjar<sup>4</sup>**

<sup>1</sup>*University of Technology of Belfort-Montbéliard, 90010 Belfort, France*

<sup>2</sup>*University of the Basque Country, 20018 San Sebastián, Spain*

<sup>3</sup>*IKERBASQUE, Basque Foundation for Science, 48013 Bilbao, Spain*

<sup>4</sup>*University of Lille, 59650 Villeneuve d'Ascq, France*

Correspondence should be addressed to Yassine Ruichek; [yassine.ruichek@utbm.fr](mailto:yassine.ruichek@utbm.fr)

Received 8 June 2016; Accepted 8 June 2016

Copyright © 2016 Yassine Ruichek 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.

During the past decades, intelligent vehicles (advanced driving assistance systems and autonomous driving) and mobile robotic systems have received more and more attentions and developments from both research society and industry community. One of the necessary components to develop ADAS systems, driverless cars, or autonomous navigation mobile robots is perception of the surrounding vehicle environment. In intelligent vehicles, perception systems are able to sense and interpret surrounding environment based on various kinds of sensors, such as radar, sonar sensors, 2D/3D lidar, monocular/binocular/omnidirectional vision system, and inertial sensors. The perception systems provide and process sensed information for representing dynamically the content of the surrounding environment (detection, tracking, and recognition of static and dynamic objects). Therefore, processing and interpreting data provided by these sensors are required for making decision in applications related to driver assistance and vehicle control in complex environments. This special issue aims at exhibiting the latest research achievements, related to the following topics:

- (i) sensors calibration methods;
- (ii) sensor fusion and information integration;
- (iii) sensor-based vehicle localization in GPS denied environment;
- (iv) sensor-based localization and SLAM;
- (v) sensor-based object detection and tracking;
- (vi) sensor-based scene analysis and understanding.

This special issue received 26 papers. After reviewing process, 9 papers are accepted for publication. According to their content, the accepted papers could be classified into two classes. The first one gathers papers addressing environment perception. In this class of papers, problems discussed are camera rectification in dynamic environments, SLAM for indoor robots, map building, night-time perception, vision based vehicle speed estimation, and so forth. The second class is concerned with papers dealing with vehicle localization. The papers of this class focused on multisensory based navigations systems for cars, trains, and UAV.

*Yassine Ruichek  
Fadi Dornaika  
Maan El Badaoui El Najjar*

## Research Article

# FastSLAM Using Compressed Occupancy Grids

Christopher Cain<sup>1</sup> and Alexander Leonessa<sup>2</sup>

<sup>1</sup>*SEW-EURODRIVE, Lyman, SC 29365, USA*

<sup>2</sup>*Center for Dynamic Systems Modeling and Control, Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA 24061, USA*

Correspondence should be addressed to Alexander Leonessa; [aleoness@vt.edu](mailto:aleoness@vt.edu)

Received 21 October 2015; Accepted 8 May 2016

Academic Editor: Maan E. El Najjar

Copyright © 2016 C. Cain and A. Leonessa. 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.

Robotic vehicles working in unknown environments require the ability to determine their location while learning about obstacles located around them. In this paper a method of solving the SLAM problem that makes use of compressed occupancy grids is presented. The presented approach is an extension of the FastSLAM algorithm which stores a compressed form of the occupancy grid to reduce the amount of memory required to store the set of occupancy grids maintained by the particle filter. The performance of the algorithm is presented using experimental results obtained using a small inexpensive ground vehicle equipped with LiDAR, compass, and downward facing camera that provides the vehicle with visual odometry measurements. The presented results demonstrate that although with our approach the occupancy grid maintained by each particle uses only 40% of the data needed to store the uncompressed occupancy grid, we can still achieve almost identical results to the approach where each particle filter stores the full occupancy grid.

## 1. Introduction

One of the most important problems in the development of an unmanned vehicle (UMV) is giving it the ability to be placed in a completely unknown environment and allowing it to determine (1) its surroundings, based on what it “sees” using available sensors, and (2) its position, based on how far it has moved and what it sees. In the field of robotics this problem is referred to as Simultaneous Localization and Mapping (SLAM) or Concurrent Mapping and Localization (CML). In this paper we develop a method of solving the SLAM problem which implements a representation of the world using a compressed occupancy grid [1].

The approach we develop is an extension of the FastSLAM algorithm [2] which is one of the first solutions to the SLAM problem that makes use of a particle filter. More specifically the FastSLAM algorithm uses a Rao-Blackwellized [3, 4] particle filter to generate a position estimate of the UMV while each particle in filter maintains its copy of the occupancy grid. Based on the expected size of the environment in which a UMV operates and the level of detail required, the size of each occupancy grid can become extremely large

(matrices holding hundreds of thousands of entries). Furthermore, as discussed further in Section 3.1.2, the probability that the estimate produced by the FastSLAM algorithm is accurate increases as the number of particles in the particle filter increases. For these reasons the amount of computer resources required to perform the algorithm can grow quickly. This problem is exacerbated in small form factor and low cost UMVs as the computing resources available to vehicles of this type are much less than the resources available to larger and more expensive vehicles. The approach that we develop extends the FastSLAM algorithm [2] to use compressed occupancy grids to overcome the memory restrictions that can occur when storing full occupancy grids without making drastic changes to the original algorithm.

The remainder of this paper is organized as follows. The mathematical notations used in this paper along with some mathematical preliminaries are presented in Section 2. The updated FastSLAM algorithm that makes use of compressed occupancy grids is introduced in Section 3. In Section 4 four separate approaches for reconstructing the compressed occupancy grid are examined. Section 5 presents a set of performance improvements to some components of the

algorithm. In Section 6 experimental results are provided and concluding remarks are presented in Section 7.

## 2. Mathematical Preliminaries

In much of the existing literature, the SLAM problem is addressed in a probabilistic sense. In many cases we would like to estimate the probability, also referred to as the distribution, of some random variable  $x \in \mathbb{R}$  and we denote the distribution of  $x$  as  $p(x)$ . In many cases we use some additional information  $y \in \mathbb{R}$  to tell us something about the random variable  $x$ . In this situation  $p(x)$  is referred to as the *prior probability* and it is all that we know about the probability of  $x$  without the inclusion of  $y$ , in many cases this is shortened to *prior*. The distribution of  $x$  with the inclusion of the data  $y$  is denoted as  $p(x | y)$  and referred to as the *posterior probability* and in many cases is just referred to as the *posterior*.

In SLAM we are attempting to estimate the pose  $\xi_k$  of a UMV and a map  $\mathbf{M}_k$  of the environment that surrounds the UMV at some time step  $k > 0$ . In this paper we treat the world in which the UMV operates as a two-dimensional plane; thus  $\xi_k \triangleq [x_k \ y_k \ \theta_k]^T$ , where  $[x_k \ y_k] \in \mathbb{R}^2$  are the horizontal and vertical positions of the vehicle in some frame of reference and  $\theta_k \in (-\pi, \pi]$  is the heading of the UMV with respect to the positive horizontal axis in the frame of reference. From our use of occupancy grids, the estimated map is represented by a matrix  $\mathbf{M}_k \in \mathbb{R}^{r \times c}$ , where  $r \in \mathbb{N}$  are the number of rows in the grid and  $c \in \mathbb{N}$  are the number of columns. We assume that the environment that the UMV operates is static; therefore during the time in which the UMV is performing the SLAM algorithm the environment does not change. Based on this assumption, to simplify the notation, the occupancy grid is denoted with  $\mathbf{M}$ . The estimate produced by SLAM, in many cases, makes use of sensor measurements and control inputs. The set of sensor measurements  $\mathbf{z}_{1:k}$  denotes the full set of sensor measurements for  $k > 0$ ; thus  $\mathbf{z}_{1:k} \triangleq [\mathbf{z}_1, \dots, \mathbf{z}_k]$ , where  $\mathbf{z}_i \in \mathbb{R}^m$ ,  $i = 1, 2, \dots, k$ , and  $m \in \mathbb{N}$  is the number of measurements in  $\mathbf{z}_i$ . In the same way the full set of control inputs are defined as  $\mathbf{u}_{1:k} \triangleq [\mathbf{u}_1, \dots, \mathbf{u}_k]$ , where  $\mathbf{u}_i \in \mathbb{R}^c$ ,  $i = 1, 2, \dots, k$ , and  $c \in \mathbb{N}$  is the size of the control vector  $\mathbf{u}_i$ . With the notation defined and the required mathematical preliminaries provided, the FastSLAM algorithm that makes use of compressed occupancy grids is presented in the following section.

## 3. FastSLAM with Compressed Occupancy Grids

In general the goal of SLAM is to estimate the pose  $\xi_k$  of a UMV and the map,  $\mathbf{M}$ , of the unknown environment in which the UMV is operating at some time step  $k > 0$ . We present an extension of the FastSLAM algorithm that uses compressed occupancy grids to store the map. Before presenting our approach, an overview of the FastSLAM algorithm that uses standard occupancy grids (FastSLAM OG) is provided.

**3.1. FastSLAM with Occupancy Grids.** As opposed to many SLAM solutions that attempt to solve the online SLAM problem that is estimating the posterior that represents the pose of a UMV and the map of the environment based on a set of control inputs and sensor measurements, the FastSLAM approach attempts to estimate the full SLAM posterior. The full SLAM posterior is given as

$$p(\xi_{1:k}, \mathbf{M} | \mathbf{u}_{1:k}, \mathbf{z}_{1:k}), \quad (1)$$

which is the distribution that represents the trajectory of UMV  $\xi_{1:k}$  and the map of the environment  $\mathbf{M}$  based on the set of control inputs  $\mathbf{u}_{1:k}$  and observations  $\mathbf{z}_{1:k}$ . By estimating the full SLAM posterior, (1) can be factored, using the property of conditional independence, into a pair of simpler terms which is given by

$$p(\xi_{1:k}, \mathbf{M} | \mathbf{u}_{1:k}, \mathbf{z}_{1:k}) = p(\xi_{1:k} | \mathbf{u}_{1:k}, \mathbf{z}_{1:k}) \times p(\mathbf{M} | \xi_{1:k}, \mathbf{u}_{1:k}, \mathbf{z}_{1:k}), \quad (2)$$

where  $p(\xi_{1:k} | \mathbf{u}_{1:k}, \mathbf{z}_{1:k})$  is the distribution that represents the UMV trajectory and  $p(\mathbf{M} | \xi_{1:k}, \mathbf{u}_{1:k}, \mathbf{z}_{1:k})$  is the distribution that represents the map of the environment. The FastSLAM approach estimates the factored SLAM posterior (2) using a Rao-Blackwellized particle filter. The Rao-Blackwellized particle filter is a version of a sampling importance resampling (SIR) [5] particle filter and it is the SIR particle filter that provides the format of the FastSLAM solution. The Rao-Blackwellized particle filter estimates the distribution representing a portion of the state, which in FastSLAM is the UMV trajectory  $\xi_{1:k}$ , using a particle filter, and each particle in the particle filter maintains its own estimate of the remainder of the state, which is the map  $\mathbf{M}$ . As a result, each particle in the particle filter is composed of a UMV pose and a map, which the FastSLAM OG algorithm represents using an occupancy grid; thus the  $p$ th particle at time  $k$  is defined as

$$\mathcal{X}_k^{[p]} \triangleq [\xi_k^{[p]} \ \mathbf{M}_k^{[p]}], \quad (3)$$

where  $\xi_k^{[p]}$  is the pose estimate of the  $p$ th particle at time  $k$ ,  $\mathbf{M}_k^{[p]}$  is the occupancy grid of the environment maintained by the particle as it has been generated through time step  $k$ , and  $p \in [1, n]$  is the index corresponding to the  $p$ th particle where  $n$  is the total number of particles used in the particle filter. The FastSLAM approach is implemented in a four-step procedure based on the use of a SIR particle filter to estimate (1) and an overview of the algorithm is presented in Algorithm 1 for time  $k$ .

**3.1.1. Motion Model Update.** In the first step of the algorithm (Line (4) in Algorithm 1) a new particle pose is generated and added to a set of temporary particles,  $\overline{\mathcal{X}}_k$ . The new pose for the  $p$ th particle is generated by sampling from the probabilistic motion model of the UMV:

$$\xi_k^{[p]} \sim p(\xi_k | \xi_{k-1}^{[p]}, \mathbf{u}_k). \quad (4)$$

```

(1) procedure FASTSLAMOG( $\mathcal{X}_{k-1}, \mathbf{u}_k, \mathbf{z}_k$ )
(2)    $\overline{\mathcal{X}}_k \leftarrow \{\}$ 
(3)   for  $p \leftarrow 1, n$  do
(4)      $\xi_k^{[p]} \sim p(\xi_k | \xi_{k-1}^{[p]}, \mathbf{u}_k)$ 
(5)      $w^{[p]} \leftarrow \eta p(\mathbf{z}_k | \xi_k^{[p]}, \mathbf{M}_{k-1}^{[p]})$ 
(6)      $\mathbf{M}_k^{[p]} = \text{OGUPDATE}(\xi_k^{[p]}, \mathbf{M}_{k-1}^{[p]}, \mathbf{z}_k)$ 
(7)      $\overline{\mathcal{X}}_k \leftarrow [\xi_k^{[p]} \ \mathbf{M}_k^{[p]} \ w^{[p]}]$ 
(8)   end for
(9)    $\mathcal{X}_k \leftarrow \{\}$ 
(10)  for  $p = 1, n$  do
(11)    select  $j$  with probability proportional to  $w^{[p]}$ 
(12)     $\mathcal{X}_k^{[p]} \leftarrow \overline{\mathcal{X}}_k^{[j]}$ 
(13)  end for
(14) end procedure

```

ALGORITHM 1: FastSLAM OG algorithm.

The probabilistic motion model of the UMV generates a new pose based on the state transition model of the UMV, the pose of the  $p$ th particle at time step  $k-1$ , the current control input  $\mathbf{u}_k$ , and the characteristics of the noise on  $\mathbf{u}_k$ . The result is that the particles in  $\overline{\mathcal{X}}_k$  are distributed according to

$$p(\xi_{1:k} | \mathbf{u}_{1:k}, \mathbf{z}_{1:k-1}), \quad (5)$$

which is referred to as the *proposal distribution*. The actual distribution which we are attempting to estimate is referred to as the *target distribution* and given as

$$p(\xi_{1:k} | \mathbf{u}_{1:k}, \mathbf{z}_{1:k}). \quad (6)$$

The proposal distribution is transformed to the target distribution in the last stage of the algorithm, the resampling stage (Lines (10)–(13) in Algorithm 1). Before the resampling process can occur,  $\mathbf{z}_k$  must be incorporated into the estimate, which is achieved in the calculation of the particle's importance weight.

**3.1.2. Importance Weight Calculation.** In the second step of the algorithm an importance weight for each particle is generated. The need for an importance weight for each particle comes from the use of a SIR particle filter and, as seen in the previous section, we are sampling particles from a distribution different from the one that we are attempting to estimate. According to [6], if we are unable to directly sample from the distribution that we are attempting to estimate, then the importance weight for each particle is given as

$$w_k^{[p]} = \frac{\text{target distribution}}{\text{proposal distribution}}, \quad (7)$$

and particles are drawn from  $\overline{\mathcal{X}}_k$  with replacement and added to  $\mathcal{X}_k$  with a probability proportional to  $w_k^{[p]}$ . When drawing particles from  $\overline{\mathcal{X}}_k$  with replacement, each particle that is added to  $\mathcal{X}_k$  remains in  $\overline{\mathcal{X}}_k$  so that a given particle that is selected has the potential of being added to  $\mathcal{X}_k$  multiple

times. After the resampling process  $\mathcal{X}_k$  will approximate the target distribution and the quality of the approximation will improve, the number of particles increases.

Using (7) along with the previously discussed distributions (5) and (6), the importance weight of each particle simplifies to [7]

$$w_k^{[p]} = \frac{p(\xi_{1:k} | \mathbf{u}_{1:k}, \mathbf{z}_{1:k})}{p(\xi_{1:k} | \mathbf{u}_{1:k}, \mathbf{z}_{1:k-1})} = \eta p(\mathbf{z}_k | \xi_k^{[p]}, \mathbf{M}_{k-1}^{[p]}), \quad (8)$$

where  $p(\mathbf{z}_k | \xi_k^{[p]}, \mathbf{M}_{k-1}^{[p]})$  is the probabilistic measurement model of the sensor being used by the UMV and  $\eta > 0$  is a normalizing factor. The probabilistic measurement model is dependent on the perceptual sensor being used by the UMV and it calculates the probability that the sensor measurement would make based on the current pose estimate and map maintained by the particle. This form of importance weight incorporates the sensor observations into the estimate which were not included in the pose sampling procedure. Once the importance weight has been calculated for each particle then the occupancy grid maintained by each particle is updated using the new particle pose and current sensor measurement.

**3.1.3. Occupancy Grid Update.** Once the importance weight for each particle has been calculated, the third step of the algorithm updates the occupancy grid of each particle using the current sensor measurement and pose estimate. Occupancy grids decompose an unknown environment into a finite set of cells where each cell holds the probability that the location in the world enclosed by the cell is occupied by an object. From [8], the occupancy grid for a given particle at time  $k$  is obtained in a probabilistic sense by calculating the posterior over maps based on the history of sensor measurements and trajectory of a particle:

$$p(\mathbf{M}_k^{[p]} | \xi_{1:k}^{[p]}, \mathbf{z}_{1:k}). \quad (9)$$

By decomposing the environment into cells, the posterior calculated by the occupancy grid can be restated as a product of cellular posteriors:

$$p(\mathbf{M}_k^{[p]} | \xi_{1:k}^{[p]}, \mathbf{z}_{1:k}) = \prod_{i,j} p(\mathbf{M}_k^{[p]}(i,j) | \xi_{1:k}^{[p]}, \mathbf{z}_{1:k}), \quad (10)$$

where  $p(\mathbf{M}_k^{[p]}(i,j) | \xi_{1:k}^{[p]}, \mathbf{z}_{1:k})$  contains the probability that the location in the environment represented by the cell at  $(i,j)$  is occupied based on the measurement history and particle trajectory. A solution to a problem of this form is the binary Bayes filter [8]. The binary Bayes filter stores the probability of occupancy for each cell in log odds form to prevent truncation errors that can occur during the update process. The log odds form of the variable  $x \in \mathbb{R}$  is given as

$$l(x) \triangleq \log \frac{x}{1-x}, \quad (11)$$

and the value of  $x$  can be recovered from the log odds form as

$$x \triangleq 1 - \frac{1}{1 + \exp(l(x))}. \quad (12)$$

```

(1) procedure OGPUUPDATE( $\xi_k, \mathbf{M}_{k-1}, \mathbf{z}_k$ )
(2)   for  $i = 1, h$  do
(3)     for  $j = 1, w$  do
(4)       for  $l = 1, m$  do
(5)         if  $\mathbf{M}_{k-1}(i, j)$  lies in the perceptual range of  $\mathbf{z}_k(l)$  then
(6)            $\mathbf{M}_k(i, j) \leftarrow \mathbf{M}_{k-1}(i, j) + \log \frac{p(\mathbf{M}(i, j) | \xi_k, \mathbf{z}_k(l))}{1 - p(\mathbf{M}(i, j) | \xi_k, \mathbf{z}_k(l))} - l_0$ 
(7)         end if
(8)       end for
(9)     end for
(10)   end for
(11) end procedure

```

ALGORITHM 2: Standard occupancy grid update algorithm.

The binary Bayes filter based occupancy grid update modifies the probability of occupancy of each cell in the grid when an observation provides information about that location. As a consequence, for locations that have been observed, the probability of occupancy quickly approaches 0 for free cells and 1 for occupied cells. While the addition of new information to the estimate makes the probability quickly approach the limits, the probability of occupancy of each cell should never equal 0 or 1 as this would assume perfect knowledge about the occupancy of the cell. Depending on the data type selected to store the probability, the precision required to store the value can quickly be exceeded resulting in the probability being rounded to either 0 or 1. Once this rounding has occurred any new information learned about the occupancy of a cell is lost. To overcome this problem the binary Bayes filter stores the probability of occupancy in log odds form which takes the potential probability values in  $[0, 1]$  and spreads them over the range  $(-\infty, \infty)$ . By spreading the small range of probability values over this much larger range, the data type being used to store the probability has more precision to store the probability thus preventing the truncation/rounding from occurring. The occupancy grid update algorithm based on the binary Bayes filter is provided in Algorithm 2.

In Algorithm 2,  $p(\mathbf{M}(i, j) | \xi_k, \mathbf{z}_k(l))$  is the inverse sensor model of the sensor being used to generate the map and it returns the probability that the cell at  $(i, j)$  is occupied based on the  $l$ th sensor measurement and the pose estimate of the particle. The value of  $l_0$  in Algorithm 2 is referred to as the prior of occupancy and is defined as

$$l_0 \triangleq \log \frac{p(\mathbf{M}(i, j))}{1 - p(\mathbf{M}(i, j))} \quad (13)$$

and gives the probability that a cell is occupied before any sensor measurements are integrated into the map. Typically  $p(\mathbf{M}(i, j))$  in (13) is selected to be 0.5.

**3.1.4. Resampling.** After the occupancy grid for each particle in  $\overline{\mathcal{X}}_k$  has been updated, the final step of the algorithm is the resampling process. During the resampling process particles are drawn with replacement from  $\overline{\mathcal{X}}_k$  with a probability

proportional to  $w^{[p]}$ . This resampling step takes the set of temporary particles that are distributed according to (5) and ensures that there is a high probability that  $\overline{\mathcal{X}}_k$  is distributed according to (6).

One of the problems with this approach is the large amount of memory required to run the algorithm. The estimate generated by the algorithm improves as the number of particles increases; thus we like to have as many particles as possible. However, each particle must keep a copy of the occupancy grid and each occupancy grid can be quite large. It is easy to see that while we would like to run the particle filter with as many particles as possible we can be limited by the fact that each particle must maintain a full copy of the occupancy grid. To address this problem we propose an updated version of the algorithm that decreases the amount of memory required for each particle by storing a compressed version of the occupancy grid as opposed to the full uncompressed occupancy grid.

**3.2. FastSLAM with Compressed Occupancy Grids.** Before discussing the updated SLAM algorithm, we will first address the compressed form of the occupancy grid. To compress  $\mathbf{M}_k^{[p]}$  we first reshape the occupancy grid so that  $r \times c$  matrix becomes a column vector containing  $n = rc$  elements and we refer to the vector form of the occupancy grid as  $\mathbf{m}_k^{[p]} \triangleq \text{vec}(\mathbf{M}_k^{[p]})$ . The reshaping process is performed in such a way that the cell in the original occupancy grid located at  $(i, j)$  can be accessed in the vector form at  $l$ , where  $l = ic + j$ . In order to compress the occupancy grid without major degradation, we assume that the occupancy grid can be represented in an alternate basis in which it is sparse. The relationship between the two different representations is given as

$$\mathbf{m}_k^{[p]} = \Phi \mathbf{c}_k^{[p]}, \quad (14)$$

where  $\Phi$  is a transformation matrix that defines the relationship and  $\mathbf{c}_k^{[p]}$  is the set of coefficients that represent the occupancy grid in the alternate basis. In order for the occupancy grid to be compressible,  $\mathbf{c}_k^{[p]}$  must be sparse [9]; that is  $\#\{\text{supp}(\mathbf{c}_k^{[p]})\} < n$ , where  $\text{supp}(\cdot)$  is the support operator and  $\#$  denotes the cardinality of the set. This means

```

(1) procedure FASTSLAMCOG( $\mathcal{X}_{k-1}, \mathbf{u}_k, \mathbf{z}_k$ )
(2)    $\bar{\mathcal{X}}_k \leftarrow \{ \}$ 
(3)   for  $p \leftarrow 1, n$  do
(4)      $\xi_k^{[p]} \sim p(\xi_k | \xi_{k-1}^{[p]}, \mathbf{u}_k)$ 
(5)      $\bar{\mathbf{c}}_k^{[p]} = \text{UNCOMPRESS}(\mathbf{y}_{k-1}^{[p]})$ 
(6)      $w^{[p]} \leftarrow \eta p(\mathbf{z}_k | \xi_k^{[p]}, \Phi \bar{\mathbf{c}}_k^{[p]})$ 
(7)      $\alpha_k^{[p]} = \text{ALPHA CONSTRUCTION}(\mathbf{m}_{k-1}^{[p]}, \mathbf{z}_k)$ 
(8)      $\mathbf{y}_k^{[p]} = \mathbf{H}[\bar{\mathbf{c}}_k^{[p]} + \Phi^T[\alpha_k^{[p]} - \beta]]$ 
(9)      $\bar{\mathcal{X}}_k^{[p]} \leftarrow [\xi_k^{[p]} \quad \mathbf{y}_k^{[p]} \quad w^{[p]}]$ 
(10)   end for
(11)    $\mathcal{X}_k \leftarrow \{ \}$ 
(12)   for  $j = 1, n$  do
(13)     sample with replacement from select  $i$  with probability proportional to  $w^{[p]}$ 
(14)      $\mathcal{X}_k^{[p]} \leftarrow \bar{\mathcal{X}}_k^{[p]}$ 
(15)   end for
(16) end procedure

```

ALGORITHM 3: FastSLAM algorithm with compressed occupancy grids.

that, in order for us to be able to compress the occupancy grid, we must find a basis in which the coefficient vector is composed of a large number of zeros. If this condition is met then the occupancy grid for a given particle can be compressed using

$$\mathbf{y}_k^{[p]} = \mathbf{H}\mathbf{c}_k^{[p]}, \quad (15)$$

where  $\mathbf{H}$  is a matrix that performs the compression by selecting a subset of coefficient data that is smaller than the amount of data in the complete signal.

The FastSLAM COG algorithm is a modification of Algorithm 1. Since we are now maintaining the compressed form of the occupancy grid,  $i$ th particle in the particle set is defined as

$$\mathcal{X}_k^{[p]} \triangleq [\xi_k^{[p]} \quad \mathbf{y}_k^{[p]}]. \quad (16)$$

As with FastSLAM OG, FastSLAM COG is performed in several steps and an overview of the updated algorithm is presented in Algorithm 3. As a result of using compressed occupancy grids the only changes that must be addressed are the steps of the algorithm that require the occupancy grid, the importance weight calculation, and occupancy grid update.

**3.2.1. Importance Weight Calculation.** As discussed in Section 3.1.2, the importance weight for each particle is the normalized probabilistic measurement model for each particle. The probabilistic measurement model is a measure of how well the current set of sensor measurements matches up with what we expect based on the current pose and map of the particle. In calculating the probabilistic measurement model, the algorithm must have access to the raw occupancy grid values so that an expected sensor measurement can be generated. Based on this requirement, the compressed occupancy grid coefficients are first uncompressed. For now a general reconstruction operator denoted as  $\mathbf{H}_r$  is used;

however the details of several reconstruction methods are explored in detail in Section 4. Once the compressed coefficients have been reconstructed, the occupancy grid can be extracted using (14). The importance weight calculation can be found in Algorithm 3 on Line (6).

**3.2.2. Compressed Occupancy Grid Update.** As previously seen in Algorithm 2, the standard approach to updating the occupancy grid is to examine every cell in the occupancy grid and if a given cell falls within the perceptual range of the sensor attached to the UMV then the corresponding cell of the occupancy grid is updated according to

$$\mathbf{M}_k^{[p]}(i, j) = \mathbf{M}_{k-1}^{[p]}(i, j) + \log \frac{p(\mathbf{M}(i, j) | \xi_k^{[p]}, \mathbf{z}_k(l))}{1 - p(\mathbf{M}(i, j) | \xi_k^{[p]}, \mathbf{z}_k(l))} - l_0. \quad (17)$$

Using Algorithm 2 as our basis, we rewrite the occupancy grid update in vector form. If we assume that the full occupancy grid is stored in vector form, then the occupancy grid update can be performed using

$$\mathbf{m}_k^{[p]} = \mathbf{m}_{k-1}^{[p]} + \alpha_k^{[p]} - \beta, \quad (18)$$

where  $\alpha_k^{[p]}$  is constructed using Algorithm 4 and  $\beta$  is vector with every element containing  $l_0$ .

Algorithm 4 first initializes  $\alpha_k^{[p]}$  using  $l_0$ . This ensures that cells that do not fall within the perceptual range of the sensor remain unchanged by the algorithm. For cells that fall into the perceptual range of the sensor, their  $\alpha$  value is set as the log odds form of the inverse sensor model and it is this value that either increases or decreases the likelihood that the cell is occupied. Equation (18) updates the occupancy grid in vector form; however we need to modify it so that it updates the occupancy grid which is stored using an alternative basis.

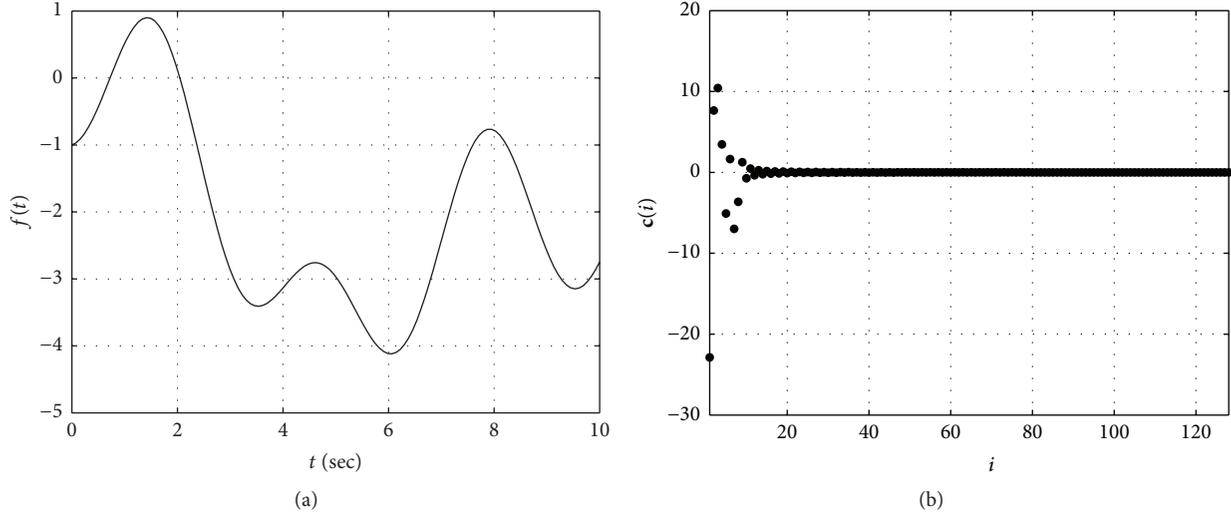


FIGURE 1: (a) The test function given by (24) and (b) the test function represented in the discrete cosine basis.

```

(1) procedure ALPHACONSTRUCTION( $\mathbf{m}, \mathbf{z}_k$ )
(2)   for  $i = 1, n$  do
(3)      $\alpha(i) = l_0$ 
(4)   end for
(5)   for  $i = 1, n$  do
(6)     for  $j = 1, m$  do
(7)       if  $\mathbf{m}(i)$  in the perceptual range of  $\mathbf{z}_k(j)$  then
(8)          $\alpha(i) = \log \frac{\mathbf{m}(i) | \xi_{1:k}^{[p]}, \mathbf{z}_k(j)}{1 - p(\mathbf{m}(i) | \xi_{1:k}^{[p]}, \mathbf{z}_k(j))}$ 
(9)       end if
(10)    end for
(11)  end for
(12) end procedure

```

ALGORITHM 4: Construction of  $\alpha^{[p]}$ .

Incorporating (14), the vector form of the occupancy grid update in the alternate basis becomes

$$\mathbf{c}_k^{[p]} = \mathbf{c}_{k-1}^{[p]} + \Phi^T \alpha_k^{[p]} - \Phi^T \beta, \quad (19)$$

where  $\Phi^T$  is used to convert  $\alpha_k^{[p]}$  and  $\beta$  to the alternate basis. In (19),  $\Phi^T$  is used to change basis as opposed to  $\Phi^{-1}$  because we assume that our choice of basis is orthonormal so  $\Phi^{-1} = \Phi^T$ ; however if the alternative basis that is selected is not orthonormal then  $\Phi^{-1}$  should replace  $\Phi^T$  in (19). We can now simplify (19) by grouping the two components that are multiplied by  $\Phi^T$  to reduce the number of matrix-vector multiplications; this yields

$$\mathbf{c}_k^{[p]} = \mathbf{c}_{k-1}^{[p]} + \Phi^T [\alpha_k^{[p]} - \beta]. \quad (20)$$

Equation (20) updates the occupancy grid in the alternate basis; hence the final step is to incorporate compression into the update. The occupancy grid maintained by each particle

contains the compressed set of basis of coefficients; thus (20) can be updated using (15) to yield

$$\mathbf{c}_k^{[p]} = \mathbf{H}_r \mathbf{y}_{k-1}^{[p]} + \Phi^T [\alpha_k^{[p]} - \beta]. \quad (21)$$

Finally, the basis coefficients used to represent the occupancy grid must be recompressed in order to be stored by the particle. Including the final compression the compressed occupancy grid update can be performed according to

$$\mathbf{y}_k^{[p]} = \mathbf{H} [\mathbf{H}_r \mathbf{y}_{k-1}^{[p]} + \Phi^T [\alpha_k^{[p]} - \beta]], \quad (22)$$

which is seen in Algorithm 3 Line (8).

#### 4. Compressed Signal Reconstruction Methods

As previously described we would like to store a compressed version of the occupancy grid as opposed to the complete occupancy grid to decrease the memory requirements of the algorithm. In order to make use of the compressed occupancy grid, a method for reconstructing the full occupancy grid from the compressed version must be selected. In this section several reconstruction methods are examined with the goal of selecting an approach to be used in FastSLAM COG. In FastSLAM COG we are attempting to compress the vector form of an occupancy grid; however in this section we will examine methods for reconstructing a generic discrete signal  $\mathbf{x} \in \mathbb{R}^n$  that has been compressed.

We represent  $\mathbf{x}$  using some alternate basis in which it is sparse and the relationship between  $\mathbf{x}$  and its sparse representation is given by

$$\mathbf{x} = \Phi \mathbf{c}, \quad (23)$$

where  $\mathbf{c} \in \mathbb{R}^n$  is a vector of coefficients that represent  $\mathbf{x}$  in the alternate basis and  $\Phi \in \mathbb{R}^{n \times n}$  is a matrix of basis functions that determines the relationship between  $\mathbf{x}$  and  $\mathbf{c}$ . The discrete signal used in this section for testing is displayed in Figure 1(a) and the set of coefficients used to represent

the signal in the discrete cosine basis [10] can be seen in Figure 1(b). The test signal is generated by evaluating

$$f(t) = \sin(t) - \cos(2t) - 3 \sin(0.25t), \quad (24)$$

at 128 evenly spaced points in  $[0, 10]$ .

If  $\mathbf{c}$  is sparse then  $\mathbf{x}$  can be compressed; this sparseness can be seen by examining Figure 1(b) and noticing that a large number of the coefficients are zero. The compression is performed by selecting a subset of the coefficient information using

$$\mathbf{y} = \mathbf{H}\mathbf{c}, \quad (25)$$

where  $\mathbf{H} \in \mathbb{R}^{m \times n}$  is the compression matrix and  $m \in \mathbb{N}$  is the size of the compressed signal with  $m < n$ .

*4.1.  $l_2$  Reconstruction ( $l_2$ ).* One of the most simple methods for reconstructing a compressed signal is to reconstruct the signal while minimizing the error between the original and reconstructed signals as measured by  $l_2$  norm, where  $l_2$  norm of a signal  $\mathbf{x} \in \mathbb{R}^n$  is defined as

$$\|\mathbf{x}\|_2 \triangleq \sqrt{\sum_{i=1}^n \mathbf{x}(i)^2}. \quad (26)$$

In examining (25), if the size of the compressed signal is the same as the original then the reconstruction can be performed according to

$$\mathbf{c} = \mathbf{H}^{-1}\mathbf{y}. \quad (27)$$

In reality the system is underdetermined due to  $m < n$ . A common solution to problems of this form uses the Moore-Penrose pseudoinverse of  $\mathbf{H}$ .

The Moore-Penrose pseudoinverse [11, 12] is used to find the solution that minimizes  $l_2$  error for systems of the form

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (28)$$

where  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{b} \in \mathbb{R}^m$ , and  $\mathbf{A} \in \mathbb{R}^{m \times n}$ . The solution to the system found using the pseudoinverse is defined as

$$\mathbf{z} = \mathbf{A}^\dagger \mathbf{b}, \quad (29)$$

where  $\mathbf{A}^\dagger$  is the pseudoinverse of  $\mathbf{A}$  and it minimizes  $l_2$  error so that

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \geq \|\mathbf{A}\mathbf{z} - \mathbf{b}\|_2, \quad (30)$$

for all  $\mathbf{x} \in \mathbb{R}^n$  if  $\mathbf{x}$  is defined as

$$\mathbf{x} = \mathbf{A}^\dagger \mathbf{b} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \mathbf{w}, \quad (31)$$

for any vector  $\mathbf{w} \in \mathbb{R}^n$ .

Using the pseudoinverse of  $\mathbf{H}$  a set of reconstructed coefficients is generated according to

$$\hat{\mathbf{c}} = \mathbf{H}^\dagger \mathbf{y}, \quad (32)$$

and the reconstructed value of the signal is generated from

$$\hat{\mathbf{x}} = \Phi \hat{\mathbf{c}}. \quad (33)$$

The results of  $l_2$  reconstruction using four separate amounts of signal information, 25%, 50%, 75%, and 95%, are presented in Figure 2. The form selected for  $\mathbf{H}$  needed to obtain such level of compression is an orthonormal Gaussian random matrix as it possesses the RIP property whose importance will be discussed in the following section.

*4.2. Compressed Sensing Reconstruction (CS).* The second reconstruction algorithm selected is based on an approach referred to in the literature as compressed sensing [13]. Compressed sensing allows for sparse signals to be reconstructed using fewer measurements that would be required according to the Nyquist/Shannon sampling principle. In compressed sensing the full set of sparse coefficients can be reconstructed from the compressed set by solving

$$\begin{aligned} \min \quad & \|\hat{\mathbf{c}}\|_0 \\ \text{subject to} \quad & \|\mathbf{y} - \mathbf{H}\hat{\mathbf{c}}\|_2^2 \leq \varepsilon, \end{aligned} \quad (34)$$

for some sufficiently small  $\varepsilon$ . In (34)  $\|\hat{\mathbf{c}}\|_0 \triangleq \#\{\text{supp}(\hat{\mathbf{c}})\}$  is a count of the number of nonzero elements in the vector  $\hat{\mathbf{c}}$ . Solving (34) is NP-hard and the solution cannot be found efficiently. A key finding of compressed sensing is that if the measurement matrix,  $\mathbf{H}$ , is constructed in such a way that it obeys the Restricted Isometry Property (RIP), which is defined as

$$(1 - \delta) \|\mathbf{c}\|_2^2 \leq \|\mathbf{H}\mathbf{c}\|_2^2 \leq (1 + \delta) \|\mathbf{c}\|_2^2, \quad (35)$$

for some small  $\delta \geq 0$ , then  $\|\cdot\|_0$  in (34) can be replaced by  $\|\cdot\|_1$  resulting in

$$\begin{aligned} \min \quad & \|\hat{\mathbf{c}}\|_1 \\ \text{subject to} \quad & \|\mathbf{y} - \mathbf{H}\hat{\mathbf{c}}\|_2^2 \leq \varepsilon. \end{aligned} \quad (36)$$

As opposed to being NP-hard to solve, (36) is a constrained convex optimization problem and can be solved efficiently using one of several available software packages with Lasso [14], NESTA [15], and l1 magic [16] being some examples. As in  $l_2$  case, once the complete set of coefficients have been estimated using the compressed set, an estimate of the original signal is generated according to (33).

In order to examine the performance of the compressed sensing reconstruction approach, the test signal is reconstructed using the same four amounts of data, 25%, 50%, 75%, and 95%, with the same orthonormal Gaussian random matrix being used to perform the compression. Equation (36) is solved using the NESTA software library and the results of the reconstruction are shown in Figure 3.

*4.3. Compressed Sensing Embedded Kalman Filtering (CSEKF).* In the previous section a reconstruction approach was introduced that allows for the reconstruction of compressed signals using less data than expected based on

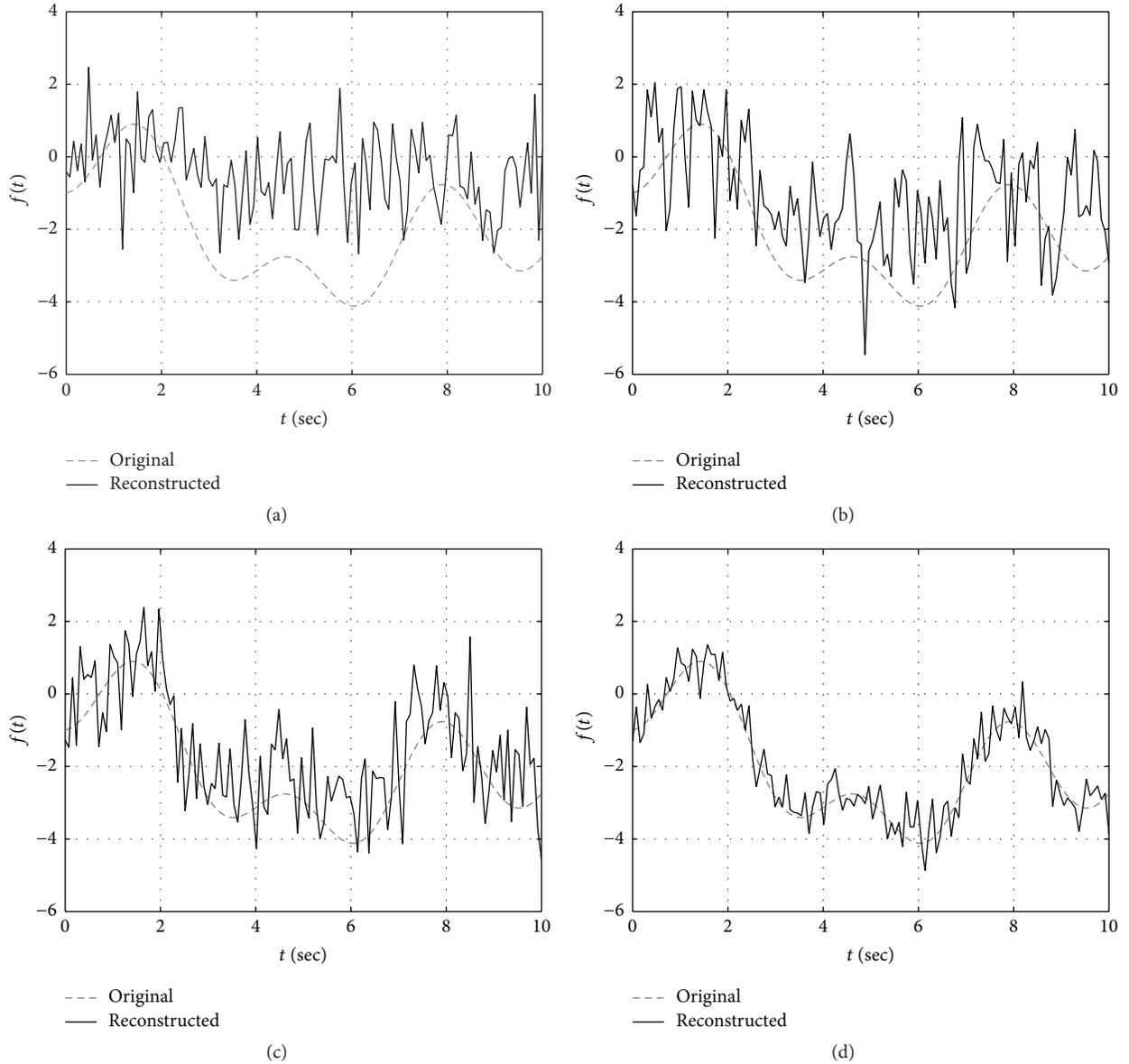


FIGURE 2: The test function reconstructed by minimizing  $l_2$  error for compressed size of (a) 25%, (b) 50%, (c) 75%, and (d) 95% of the original data.

the Nyquist/Shannon sampling principle. As seen in Figure 3 this method does a better job at reconstructing a sparse signal than the approach that reconstructs the signal by minimizing  $l_2$  norm. The authors of [17] introduce an approach that performs the compressed sensing reconstructing using an augmented Kalman filter (KF) [18] which is easy to implement, compared to the compressed sensing approach. The compressed sensing approach must solve a complex convex minimization problem, where problems of this type are typically being solved using an external software package. In order to use the KF it is assumed that the full set of coefficients evolve from time step to time step according to

$$\mathbf{c}_k = \mathbf{A}_k \mathbf{c}_{k-1} + \boldsymbol{\varepsilon}_k, \quad (37)$$

where  $\mathbf{A}_k \in \mathbb{R}^{n \times n}$  is the state transition matrix of the system which describes how the signal coefficients change between time steps, and  $\boldsymbol{\varepsilon}_k \in \mathbb{R}^n$  is a zero mean Gaussian vector with covariance  $\mathbf{R}_k \geq 0$ . The compressed form of the coefficients is generated according to

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{c}_k + \boldsymbol{\delta}_k, \quad (38)$$

where  $\mathbf{H}_k \in \mathbb{R}^{m \times n}$  is the same compression matrix discussed in (25) and  $\boldsymbol{\delta}_k \in \mathbb{R}^m$  is a zero mean Gaussian vector with covariance  $\mathbf{Q}_k > 0$ .

The CSEKF algorithm first performs a standard KF estimation that is carried out in a two parts, a *prediction* phase and a *correction* phase. The KF estimates  $p(\mathbf{c}_k | \mathbf{y}_k)$  as a Gaussian distribution that can be represented using a mean

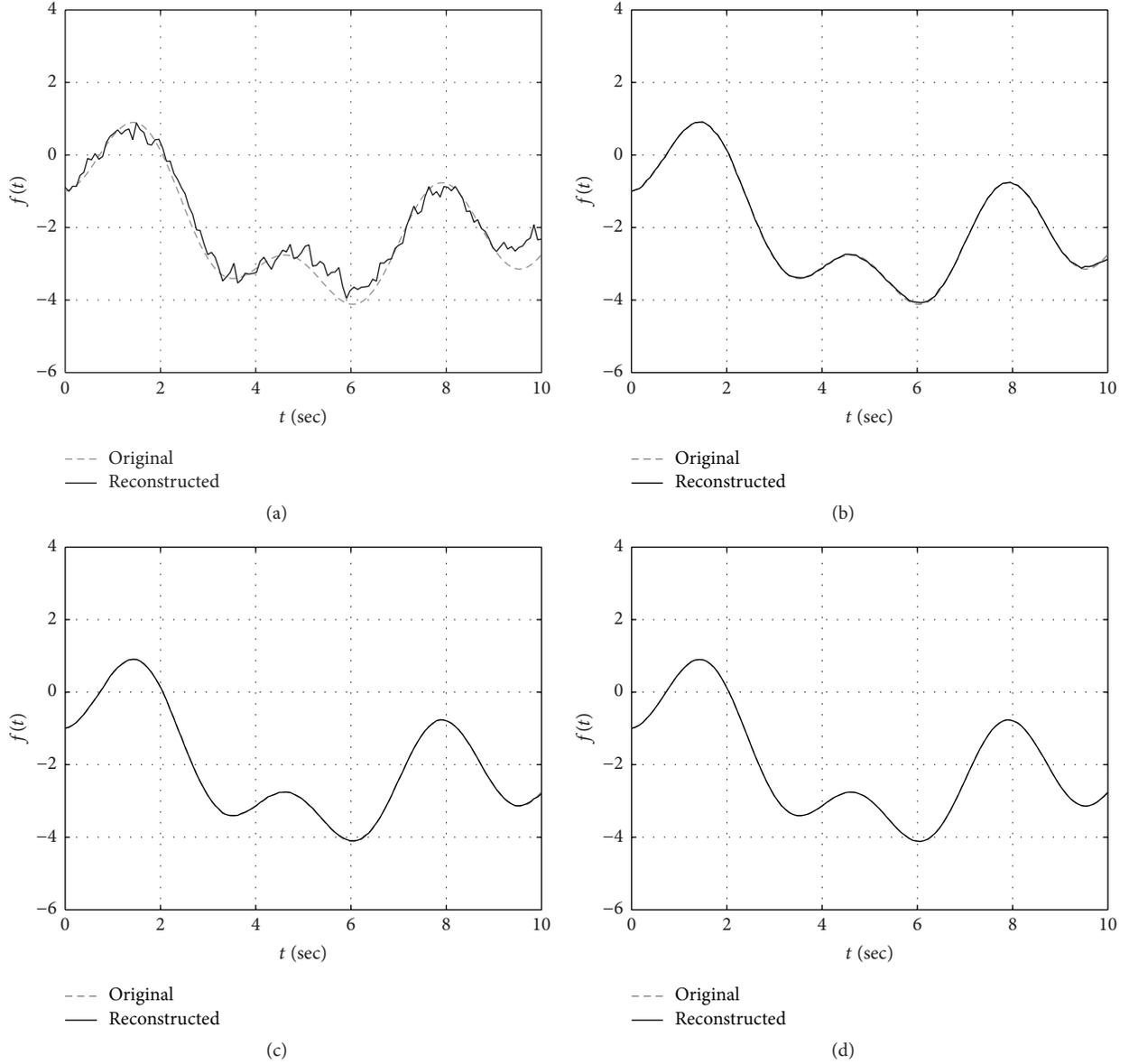


FIGURE 3: The test function reconstructed by solving the compressed sensing convex optimization problem for compressed size of (a) 25%, (b) 50%, (c) 75%, and (d) 95% of the original data.

vector  $\hat{\mathbf{c}}_k$  and covariance matrix  $\Sigma_k$ . The KF generates the estimate using a standard set of five equations [18]:

$$\bar{\mathbf{c}}_k = \mathbf{A}_k \hat{\mathbf{c}}_{k-1}, \quad (39)$$

$$\bar{\Sigma}_k = \mathbf{A}_k \Sigma_{k-1} \mathbf{A}_k^T + \mathbf{R}_k, \quad (40)$$

$$\mathbf{K}_k = \bar{\Sigma}_k \mathbf{H}_k^T (\mathbf{H}_k \bar{\Sigma}_k \mathbf{H}_k^T + \mathbf{Q}_k)^{-1}, \quad (41)$$

$$\hat{\mathbf{c}}_k = \bar{\mathbf{c}}_k + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \bar{\mathbf{c}}_k), \quad (42)$$

$$\Sigma_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \bar{\Sigma}_k, \quad (43)$$

where  $\mathbf{I}$  in (43) is  $m$  dimensional identity matrix. The estimate produced by the KF minimizes the error, using  $l_2$  norm, between the estimate and the actual set of coefficients

$$\min \|\mathbf{c}_k - \hat{\mathbf{c}}_k\|_2^2; \quad (44)$$

however, this is not what we are trying to solve according to the compressed sensing approach. The CSEKF presented by the authors replaces the classic compressed sensing convex optimization problem (36) with the dual problem:

$$\begin{aligned} \min \quad & \|\mathbf{c}_k - \hat{\mathbf{c}}_k\|_2^2 \\ \text{subject to} \quad & \|\hat{\mathbf{c}}_k\|_1 \leq \varepsilon'. \end{aligned} \quad (45)$$

They solve this constrained optimization problem by iteratively applying the pseudoobservation method presented

in [19] to the estimate generated by the KF. According to this approach a pseudoobservation is generated using the constraint being applied to the estimate which is defined as

$$\hat{\mathbf{z}}_k = \|\hat{\mathbf{c}}_k\|_1 - \varepsilon', \quad (46)$$

where the constraining value  $\varepsilon'$  is now treated as a Gaussian noise on the pseudomeasurement with a covariance of  $\mathbf{P}$ . Using the definition of  $\|\cdot\|_1$  the pseudomeasurement can be written as

$$\hat{\mathbf{z}}_k = \bar{\mathbf{H}}_k \hat{\mathbf{c}}_k - \varepsilon', \quad (47)$$

where the observation matrix of the pseudoobservation is defined as

$$\bar{\mathbf{H}} = [\text{sign}(\hat{\mathbf{c}}_k(1)) \cdots \text{sign}(\hat{\mathbf{c}}_k(n))],$$

$$\text{sign}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & \text{otherwise.} \end{cases} \quad (48)$$

The constraint is then iteratively applied  $\tau$  times to the estimate produced by the KF using the standard KF correction equations (41)–(43). First, a new Kalman matrix is generated using the observation matrix of the pseudomeasurement and the covariance of the constraint:

$$\bar{\mathbf{K}} = \Sigma_k \bar{\mathbf{H}}_k^T (\bar{\mathbf{H}}_k \Sigma_k \bar{\mathbf{H}}_k^T + \mathbf{P})^{-1}. \quad (49)$$

Next, the mean of the estimate is updated using the standard mean update equation with the assumption that the measured value is 0. Using this assumption the mean update equation can be simplified to

$$\hat{\mathbf{c}}_k = (\mathbf{I} - \bar{\mathbf{K}} \bar{\mathbf{H}}) \hat{\mathbf{c}}_k. \quad (50)$$

Finally, the covariance of the estimate is constrained using the standard covariance measurement update equation

$$\Sigma_k = (\mathbf{I} - \bar{\mathbf{K}} \bar{\mathbf{H}}_k) \Sigma_k. \quad (51)$$

The full algorithm that combines all of these separate components to reconstruct the compressed signal using the CSEKF is shown in Algorithm 5.

The reconstructed test function for the same four test percentages as in the previous two algorithms is shown in Figure 4. The algorithm is implemented as shown in Algorithm 5 with the tuning parameters being chosen as  $\mathbf{Q}_k = 0.01^2$ ,  $\mathbf{R}_k = 0$ ,  $\mathbf{P} = 100^2$ , and  $\tau = 1000$ . Since the input vector is static the initial estimate of the coefficient vector was zero,  $\hat{\mathbf{c}}_{k-1} = 0^n$ , the state transition matrix is chosen as  $\mathbf{A}_k = \mathbf{I}^n$ , and the initial covariance matrix of the estimate is chosen to be  $\Sigma_{k-1} = \mathbf{I}^n$ .

*4.4. Kalman Filter Constrained with Quasinorm (CSEKF- $p$ ).* In the previous section an approach was presented that solves a problem similar to the standard compressed sensing problem (36) using a KF that is constrained by iteratively applying a pseudoobservation to the estimate generated by

```

(1) procedure CSEKF( $\hat{\mathbf{c}}_{k-1}, \Sigma_{k-1}, \mathbf{A}_k, \mathbf{R}_k, \mathbf{H}_k, \mathbf{Q}_k, \mathbf{y}, \tau, \mathbf{P}$ )
(2)    $\hat{\mathbf{c}}_k \leftarrow \mathbf{A}_k \hat{\mathbf{c}}_{k-1}$ 
(3)    $\Sigma_k \leftarrow \mathbf{A}_k \Sigma_{k-1} \mathbf{A}_k^T + \mathbf{R}_k$ 
(4)    $\mathbf{K} = \Sigma_k \mathbf{H}^T (\mathbf{H}_k \Sigma_k \mathbf{H}_k^T + \mathbf{Q}_k)^{-1}$ 
(5)    $\hat{\mathbf{c}}_k + \mathbf{K}(\mathbf{y} - \mathbf{H}_k \hat{\mathbf{c}}_k)$ 
(6)   for  $i = 1, \tau - 1$  do
(7)      $\bar{\mathbf{H}} = [\text{sign}(\hat{\mathbf{c}}_k(1)) \cdots \text{sign}(\hat{\mathbf{c}}_k(n))]$ 
(8)      $\bar{\mathbf{K}} = \Sigma_k \bar{\mathbf{H}}^T (\bar{\mathbf{H}} \Sigma_k \bar{\mathbf{H}}^T + \mathbf{P})^{-1}$ 
(9)      $\hat{\mathbf{c}}_k = (\mathbf{I} - \bar{\mathbf{K}} \bar{\mathbf{H}}) \hat{\mathbf{c}}_k$ 
(10)     $\Sigma_k = (\mathbf{I} - \bar{\mathbf{K}} \bar{\mathbf{H}}) \Sigma_k$ 
(11)   end for
(12) end procedure

```

ALGORITHM 5: Compressed sensing embedded Kalman filtering reconstruction algorithm.

the KF. The standard approach to solving the compressed sensing problem replaces  $\|\cdot\|_0$  in (34) with  $\|\cdot\|_1$ . A new approach has been presented by the authors of [20] that replaces the zero norm with  $p$ -norm  $\|\cdot\|_p$  which for  $0 < p < 1$  is defined as

$$\|\mathbf{x}\|_p = \left( \sum_{i=1}^n |\mathbf{x}(i)|^p \right)^{1/p}. \quad (52)$$

This approach has been shown to yield better accuracy than using  $\|\cdot\|_1$  in some cases. Using this approach, we would like to constrain the estimate produced by the KF with the quasinorm  $\|\cdot\|_p$  as opposed to  $\|\cdot\|_1$ .

Just as in the previous section the initial estimate is produced using the standard KF equations. The pseudoobservation used to apply the constraint to the KF estimate using  $\|\cdot\|_p$  is defined as

$$\hat{\mathbf{z}}_k = \|\hat{\mathbf{c}}_k\|_p - \varepsilon', \quad (53)$$

where  $\varepsilon'$  is again treated as noise on the pseudoobservation with a covariance of  $\mathbf{P}$ . Unlike in the previous case where the constraint was linear and could be applied using the pseudoobservation matrix, the constraint applied by  $p$ -norm is nonlinear and can be rewritten as

$$\hat{\mathbf{z}}_k = \mathbf{h}(\hat{\mathbf{c}}_k) - \varepsilon', \quad (54)$$

where

$$\mathbf{h}(\hat{\mathbf{c}}_k) = \|\hat{\mathbf{c}}_k\|_p. \quad (55)$$

Since the constraint using  $p$ -norm is nonlinear the constraint is applied to the KF estimate by iteratively performing a measurement update using (54) and the Extended Kalman Filter (EKF) [21] update equations. Just as in the linear case the first step in applying the constraint is to construct the Kalman matrix according to

$$\bar{\mathbf{K}} = \Sigma_k \bar{\mathbf{H}}_k^T (\bar{\mathbf{H}}_k \Sigma_k \bar{\mathbf{H}}_k^T + \mathbf{P})^{-1}, \quad (56)$$

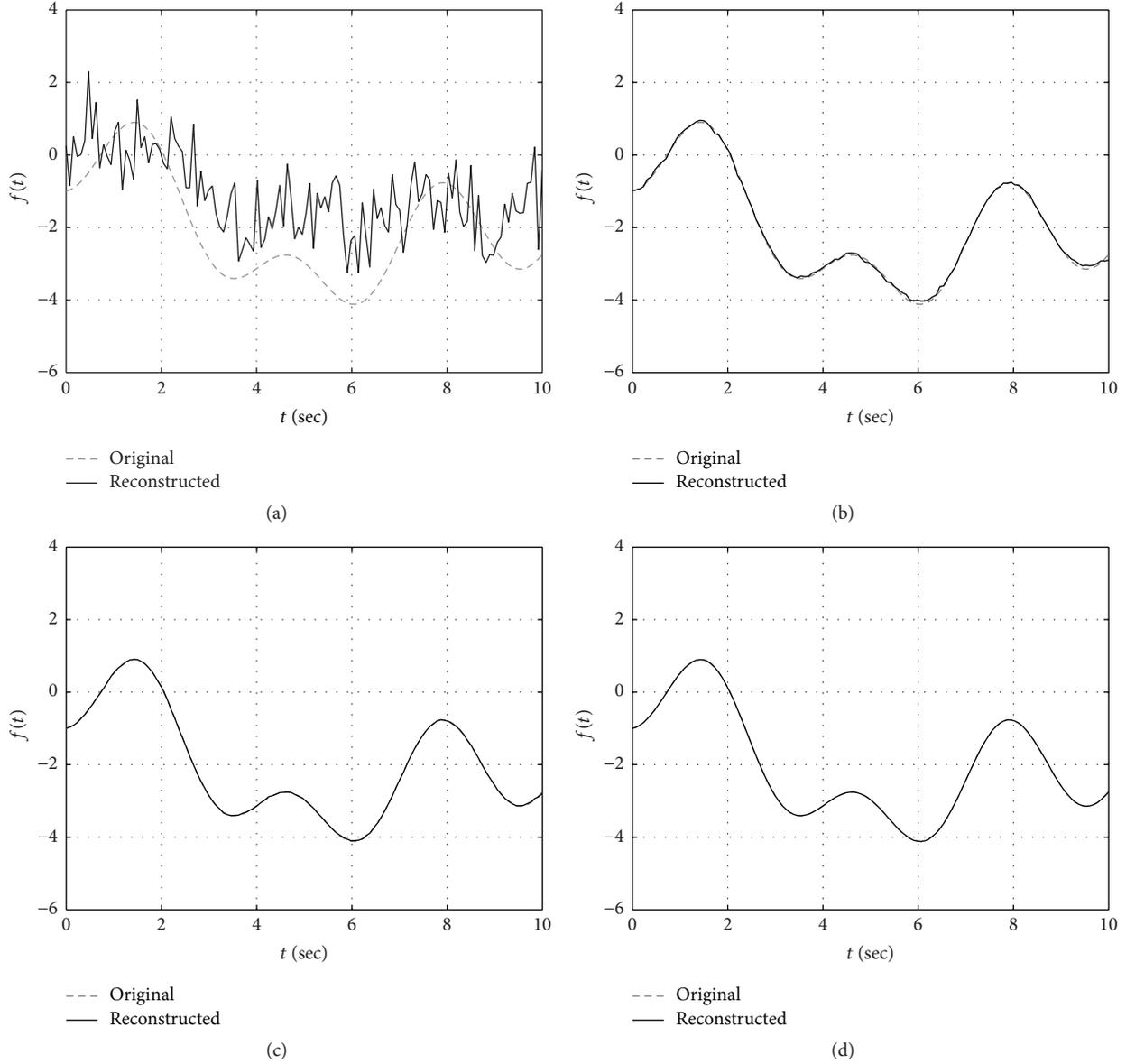


FIGURE 4: The test function reconstructed using the compressed sensing Kalman filter for compressed size of (a) 25%, (b) 50%, (c) 75%, and (d) 95% of the original data.

where  $\bar{\mathbf{H}}$  is now the Jacobian of  $\mathbf{h}(\cdot)$  and is defined as

$$\bar{\mathbf{H}}(i) = \begin{cases} \left( \sum_{i=1}^n |\hat{\mathbf{c}}_k^*(i)|^p \right)^{1/(p-1)} |\hat{\mathbf{c}}_k^*(i)|^{p-1}, & \hat{\mathbf{c}}_k^*(i) \geq 0, \\ - \left( \sum_{i=1}^n |\hat{\mathbf{c}}_k^*(i)|^p \right)^{1/(p-1)} |\hat{\mathbf{c}}_k^*(i)|^{p-1}, & \hat{\mathbf{c}}_k^*(i) < 0, \end{cases} \quad (57)$$

where  $\hat{\mathbf{c}}_k^* = \hat{\mathbf{c}}_k$ . The constraint is applied to the mean of the estimate using the standard EKF mean measurement update equation, again with the assumption that the measured value is zero, which when simplified is written as

$$\hat{\mathbf{c}}_k = \hat{\mathbf{c}}_k - \bar{\mathbf{K}} \|\hat{\mathbf{c}}_k\|_p. \quad (58)$$

The final step in applying the constraint is to update the covariance of the estimate using the standard EKF covariance measurement update equation

$$\Sigma_k = (\mathbf{I} - \bar{\mathbf{K}} \bar{\mathbf{H}}_k) \Sigma_k. \quad (59)$$

The complete algorithm for reconstructing a compressed signal using the estimate produced by the KF constrained using  $p$ -norm is shown in Algorithm 6.

The reconstruction results for the compressed sensing Kalman filter using a quasinorm to constrain the estimate are presented in Figure 5 for the same four data sizes used for the three previous algorithms. A parameter that greatly affects the performance of the algorithm is the value chosen for  $p$ . To determine the best value of  $p$  the algorithm was

```

(1) procedure CSEKFP( $\hat{\mathbf{c}}_{k-1}, \Sigma_{k-1}, \mathbf{A}_k, \mathbf{R}_k, \mathbf{H}_k, \mathbf{Q}_k, \mathbf{y}, \tau, \mathbf{P}, p$ )
(2)    $\hat{\mathbf{c}}_k \leftarrow \mathbf{A}_k \hat{\mathbf{c}}_{k-1}$ 
(3)    $\Sigma_k \leftarrow \mathbf{A}_k \Sigma_{k-1} \mathbf{A}_k^T + \mathbf{R}_k$ 
(4)    $\mathbf{K} = \Sigma_k \mathbf{H}^T (\mathbf{H}_k \Sigma_k \mathbf{H}_k^T + \mathbf{Q}_k)^{-1}$ 
(5)    $\hat{\mathbf{c}}_k + \mathbf{K}(\mathbf{y} - \mathbf{H}_k \hat{\mathbf{c}}_k)$ 
(6)   for  $i = 1, \tau - 1$  do
(7)     for  $j = 1, n$  do
(8)       if  $\hat{c}_k(j) \geq 0$  then
(9)          $\bar{\mathbf{H}}(j) = (\sum_{i=1}^n |\hat{c}_k^*(i)|^p)^{1/(p-1)} |\hat{c}_k^*(j)|^{p-1}$ 
(10)        else
(11)           $\bar{\mathbf{H}}(i) = -(\sum_{i=1}^n |\hat{c}_k^*(i)|^p)^{1/(p-1)} |\hat{c}_k^*(i)|^{p-1}$ 
(12)        end if
(13)      end for
(14)       $\bar{\mathbf{H}} = [\text{sign}(\hat{c}_k(1)) \cdots \text{sign}(\hat{c}_k(n))]$ 
(15)       $\bar{\mathbf{K}} = \Sigma_k \bar{\mathbf{H}}^T (\bar{\mathbf{H}} \Sigma_k \bar{\mathbf{H}}^T + \mathbf{P})^{-1}$ 
(16)       $\hat{\mathbf{c}}_k = (\mathbf{I} - \bar{\mathbf{K}} \bar{\mathbf{H}}) \hat{\mathbf{c}}_k$ 
(17)       $\Sigma_k = (\mathbf{I} - \bar{\mathbf{K}} \bar{\mathbf{H}}) \Sigma_k$ 
(18)    end for
(19) end procedure

```

ALGORITHM 6: Sparse signal reconstruction algorithm using the KF constrained using the quasinorm.

run for a single data percentage, 75%, for 20 equally spaced values of  $p \in (0, 1)$ . A plot of the mean square error (MSE) between the compressed signal and original signal can be seen in Figure 6. From this graph the value of  $p$  was chosen to be 0.4. The remainder of the tuning parameters for the algorithm were selected to be the same as those used for the CSEKF reconstruction.

**4.5. Algorithm Comparison.** Before deciding which reconstruction algorithm should be used in the FastSLAM COG algorithm we examine the performance of each of the algorithms in two key areas: accuracy and run time. First, we examine how well each algorithm reproduces the original signal based on the amount of information that is stored in the compressed signal. To compare how well each algorithm reconstructs the signal, each algorithm was run for 20 equally spaced percentages in (0%, 100%]. The results for each of the algorithms are shown in Figure 7 which plots the MSE as a function of the amount of data stored. As seen in the results, in general each of the compressed sensing type approaches produces less of an error for a given data percentage than  $l_2$  approach.

The second key characteristic of each algorithm examined is the time it takes for each algorithm to run. During the error performance testing described above, the time taken to run each algorithm was recorded. The run time as a function of the amount of compressed data used to represent the signal can be seen in Figure 8. As seen in the timing results the two algorithms that use the augmented Kalman filter have run times that are significantly larger than the other approaches. This increased run time is due to the number of matrix-matrix multiplications required by the Kalman filter which are computationally heavy, from [22] having  $\mathcal{O}(n^{\log_2 7})$ . The remaining two algorithms,  $l_2$  and compressed sensing,

can be implemented completely using only matrix-vector multiplications, which can be performed more quickly than matrix-matrix multiplications.

## 5. Performance Improvements

Our overall goal is to augment the FastSLAM algorithm to use compressed occupancy grids. In order for us to accomplish this task the reconstruction approach selected must run thousands or hundreds of thousands of times during the operation of a UMV. By using a Rao-Blackwellized particle filter, we can increase the probability that our SLAM posterior estimate matches the true SLAM posterior by increasing the number of particles maintained by the particle filter. As seen in Algorithm 3 the reconstruction process must be performed by each particle at each iteration; thus the execution time of the reconstruction process is important in the overall performance of the algorithm.

In Section 4 four different algorithms were presented that can be used to reconstruct an occupancy grid from the compressed form. From the timing results (Figure 8), the two approaches that make use of the Kalman filter (CSEKF and CSEKF- $p$ ) take much longer time to run than  $l_2$  and CS. The longer run times of the Kalman filter based approaches are due to the large number of matrix-matrix multiplications. Based on their performance the CSEKF and CSEKF- $p$  algorithms will not be used in FastSLAM COG. We will now present a method that can be used to improve the accuracy of the remaining two algorithms and a method that improves the run time of  $l_2$  approach.

**5.1. Updated Compression Step.** As seen in Section 3.2 the compressed occupancy grid is generated by selecting a subset

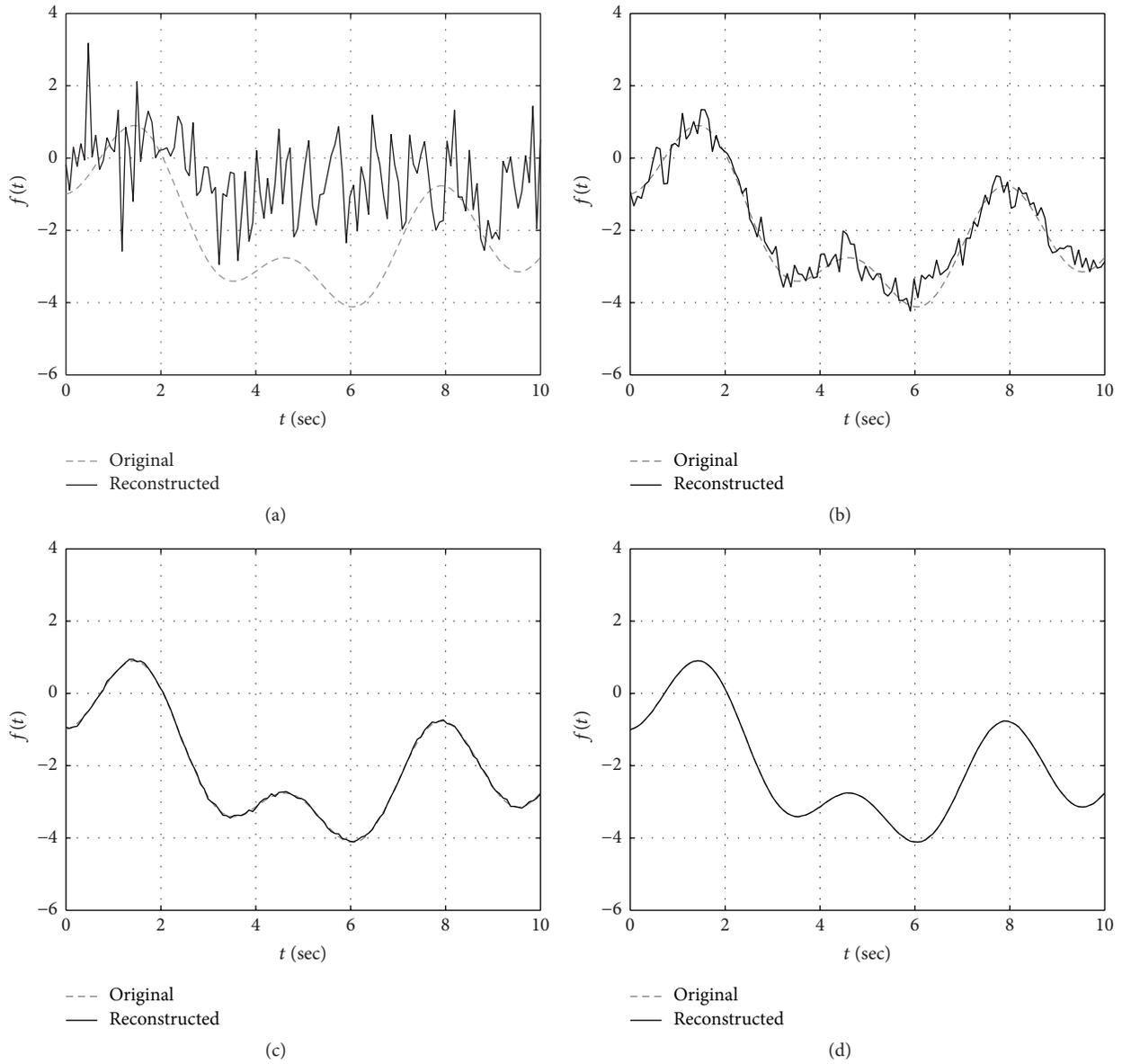


FIGURE 5: The test function reconstructed using the quasinorm constrained Kalman filter for compressed size of (a) 25%, (b) 50%, (c) 75%, and (d) 95% of the original data.

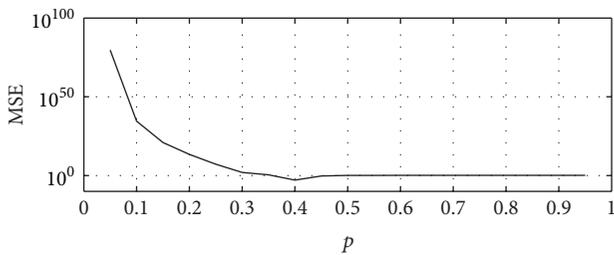


FIGURE 6: MSE for different values of  $p$  for a single compressed percentage 75%.

of the information stored in the coefficients. This compression is performed according to (15) which makes use of  $\mathbf{H}$  to perform the compression. As stated in Section 4.2 we would like for this compression matrix to obey the RIP property.

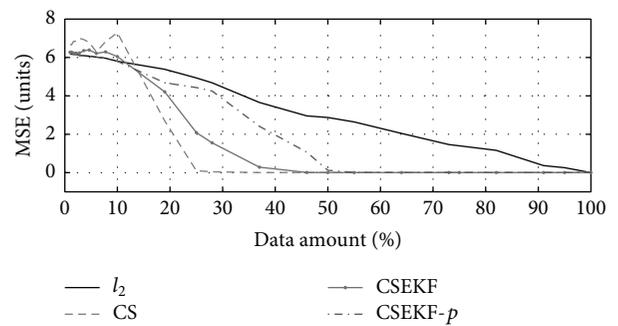


FIGURE 7: MSE as a function of signal data used for each of the four reconstruction algorithms.

Several common matrices that are used in compressed sensing applications are the discrete Fourier transform [23],

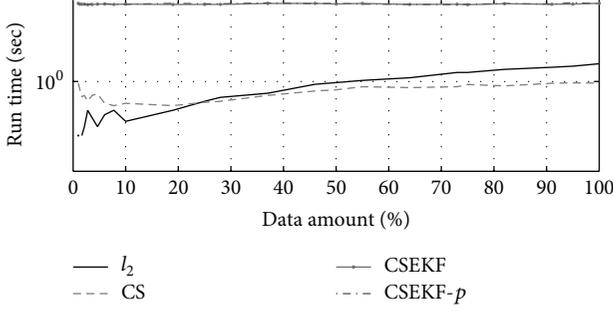


FIGURE 8: Run time as a function of compressed percentage for each of the four reconstruction algorithms.

the discrete cosine transform [10], the Hadamard transform [24], and noiselet transform [25]. Each of these matrices is constructed in such a way that each of the elements stored in the compressed vector stores a small amount of information about each of the coefficients used to represent the original signal and this information can be recovered by knowing  $\mathbf{H}$ . These types of matrices are useful when there is no knowledge about the sparse signal that is being compressed; however we are constructing the sparse signal thus we have full knowledge of the signal before the compression is performed. Using this knowledge we present an alternative form of  $\mathbf{H}$  that allows for better reconstruction using a smaller amount of information. The form of  $\mathbf{H}$  selected comes from the definition of the square error defined according to  $l_2$  norm. For some discrete signal  $\mathbf{x} \in \mathbb{R}^n$  the square of the error between the signal and the approximation of the signal  $\hat{\mathbf{x}}$  generated using  $m < n$  elements is defined as

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 = \langle \mathbf{x} - \hat{\mathbf{x}} | \mathbf{x} - \hat{\mathbf{x}} \rangle, \quad (60)$$

where  $\langle \cdot | \cdot \rangle$  is the standard inner product. The discrete signal  $\mathbf{x} \in \mathbb{R}^n$  can be represented in some alternate basis so that a single element can be expressed as

$$\mathbf{x}(i) = \sum_{j=1}^n \Phi(i, j) \mathbf{c}(j). \quad (61)$$

Recalling that  $\hat{\mathbf{y}} \in \mathbb{R}^m$  denotes the compressed set of coefficients (25), a single element of the compressed signal can be expressed in the alternate basis as

$$\hat{\mathbf{x}}(i) = \sum_{j=1}^m \hat{\Phi}(i, j) \hat{\mathbf{c}}(j), \quad (62)$$

where

$$\begin{aligned} \hat{\Phi} &= \Phi \mathbf{H}^+, \\ \hat{\mathbf{c}} &= \mathbf{y}. \end{aligned} \quad (63)$$

While this is a valid approach for compressing a signal, we would like to simplify the compression process for performance reasons. Instead of generating a new set of basis function it is easier to keep the basis functions the same at all

times and just select a subset of the coefficients to represent the signal. In order to decide which coefficients should be selected that minimize the error based on  $l_2$  norm, we define several sets. The set  $\zeta = \{x \mid x \in \mathbb{N}, 1 \leq x \leq n\}$  is the full set of indices of the discrete vector being compressed. The set  $\lambda \in \mathbb{N}^m$  is the set of indices that are used to represent the compressed signal,  $\lambda \subset \zeta$ . Finally, the set  $\gamma \in \mathbb{N}^{n-m}$  is the set of indices of coefficients not used in the compressed signal,  $\gamma = \{x \mid x \in \zeta \text{ and } x \notin \lambda\}$ . Using these sets and the fact that a constant basis is used to represent the uncompressed and compressed signal an element in the reconstructed signal can be written as

$$\hat{\mathbf{x}}(i) = \sum_{j=1}^m \Phi(i, \lambda(j)) \mathbf{c}(\lambda(j)). \quad (64)$$

The square of the error in  $l_2$  norm for a single element becomes

$$\begin{aligned} \|\mathbf{x}(i) - \hat{\mathbf{x}}(i)\|_2^2 &= \langle \mathbf{x}(i) - \hat{\mathbf{x}}(i) | \mathbf{x}(i) - \hat{\mathbf{x}}(i) \rangle \\ &= \left\langle \sum_{j=1}^{n-m} \Phi(i, \gamma(j)) \mathbf{c}(\gamma(j)) \right. \\ &\quad \left. \sum_{k=1}^{n-m} \Phi(i, \gamma(k)) \mathbf{c}(\gamma(k)) \right\rangle = \sum_{j=1}^{n-m} \sum_{k=1}^{n-m} \mathbf{c}(\gamma(j)) \mathbf{c}(\gamma(k)) \\ &\quad \times \langle \Phi(i, \gamma(j)) | \Phi(i, \gamma(k)) \rangle = \sum_{j=1}^{n-m} \Phi(i, \gamma(j))^2 \\ &\quad \cdot \mathbf{c}(\gamma(j))^2, \end{aligned} \quad (65)$$

and if we assume that the set of basis function is orthonormal, which is common, then  $l_2$  error for the entire compressed signal becomes

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 = \sum_{i=1}^n \sum_{j=1}^{n-m} \mathbf{c}(\gamma(j))^2. \quad (66)$$

This tells us that the best way to minimize the error in the reconstructed signal is to select the coefficients that have the largest magnitudes. Using this knowledge a new compression algorithm that does not require matrix-vector multiplications and selects only  $m$  coefficients with the largest magnitudes was developed and shown in Algorithm 7.

To examine how the use of the new compression algorithms affects the performance of the reconstruction algorithms, the occupancy grid shown in Figure 9(a) is reconstructed from the compressed form. The compression matrix selected for the baseline results is different from the orthonormal Gaussian random matrix used in Section 4; in its place a normalized Hadamard matrix was used. Each of the two algorithms was then run using each of the two compression methods for 20 separate compression percentages in (0, 100%). The performance results of each of the two algorithms are provided in Figure 10. As seen from the results for each of the two reconstruction algorithms the occupancy grid is reconstructed with a smaller MSE using the

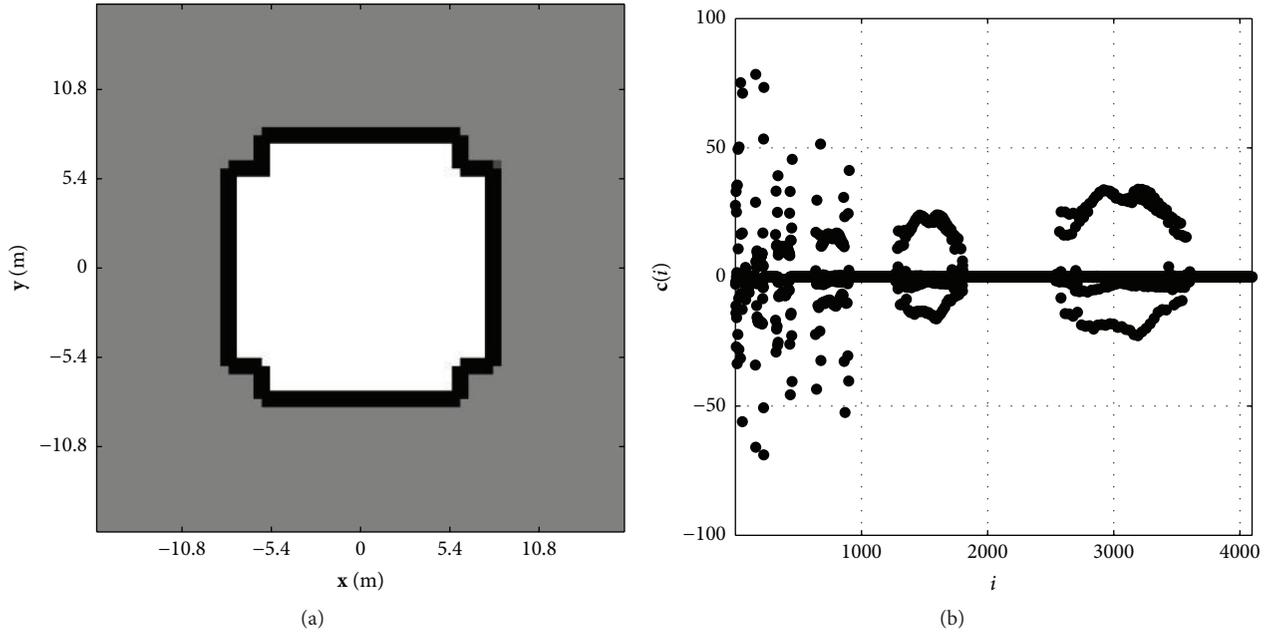


FIGURE 9: An example occupancy grid for an unknown environment (a) and the set of coefficients used to represent the occupancy grid when represented using the Haar wavelet basis (b).

```

(1) procedure FASTCOMPRESSION( $\mathbf{c}, m$ )
(2)    $\mathbf{x} = \text{SORT}(\text{abs}(\mathbf{c}), \text{"descend"})$ 
(3)   for  $i = 1, m$  do
(4)      $\mathbf{i}(i) = \mathbf{x}(i)$ 
(5)      $\mathbf{y}(i) = \mathbf{c}(\mathbf{x}(i))$ 
(6)   end for
(7) end procedure

```

ALGORITHM 7: Updated compression algorithm that represents the compressed occupancy grids with a set of coefficients and corresponding indices.

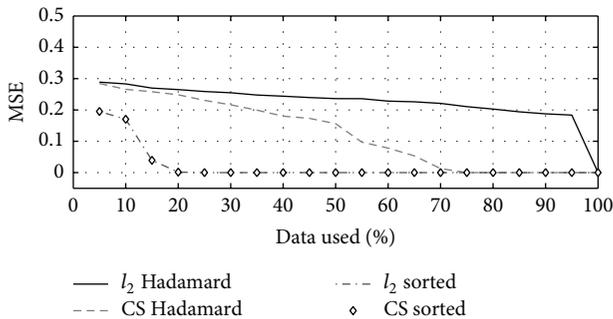


FIGURE 10: Performance results of each of the reconstruction methods using a normalized Hadamard matrix and a selection matrix to keep  $m$  most significant coefficients.

compression matrix that keeps  $m$  most significant coefficients as opposed to the compression step using the normalized Hadamard matrix using the same amount of data.

5.2. *Updated  $l_2$  Reconstruction.* In Section 4.1 a method for reconstructing a compressed signal was introduced that minimizes the error according to  $l_2$  norm. The method makes use of the Moore-Penrose pseudoinverse to reconstruct the set of coefficients from their compressed form according to

$$\hat{\mathbf{c}} = \mathbf{H}^\dagger \mathbf{y}. \quad (67)$$

This method of reconstructing the signal is made up of two steps where the performance of each step is based on the size of the signal ( $n$ ). The first step in the process is the calculation of the pseudoinverse of the compression matrix  $\mathbf{H}$ . Many common software implementations of this calculation, for example, Matlab's `pinv()` function [26], makes use of the Singular Value Decomposition (SVD) of the matrix. The process of calculating the SVD of a matrix is worse from a performance perspective than that of matrix-matrix multiplication which is  $\mathcal{O}(mn^2)$  [27] for  $m \times n$  matrix.

The second step of the algorithm is a matrix-vector multiplication operation which in a naive implementation has a performance of  $\mathcal{O}(n^2)$ . We would like to improve the performance of the reconstruction algorithm to get rid of these computationally intensive steps. By making use of the compression matrix developed in the previous section that just selects  $m$  coefficients with the largest magnitudes, the performance of the reconstruction process can be improved. Each of the rows in the compression matrix  $\mathbf{H}$  is orthogonal; that is, for two rows in the selection matrices  $\mathbf{H}(i,:)$  and  $\mathbf{H}(j,:)$  the discrete inner product is zero;  $\langle \mathbf{H}(i,:) | \mathbf{H}(j,:) \rangle = 0$ . A second property is that each row of  $\mathbf{H}$  is normalized;  $\sum_{j=1}^n \mathbf{H}(i,j) = 1$ ; thus  $\mathbf{H}$  is orthonormal. The pseudoinverse

```

(1) procedure FASTLTWORECONSTRUCTION( $\mathbf{c}, m$ )
(2)   for  $i = 1, n$  do
(3)      $\mathbf{c}(i) = 0$ 
(4)   end for
(5)   for  $i = 1, m$  do
(6)      $\mathbf{c}(\mathbf{i}(i)) = \mathbf{y}(i)$ 
(7)   end for
(8) end procedure

```

ALGORITHM 8: Updated form of  $l_2$  reconstruction algorithm that is performed without needing to calculate the pseudoinverse of the compression matrix or perform the matrix-vector multiplication.

of a orthonormal matrix can be easily computed without the computationally intensive SVD process according to

$$\mathbf{H}^\dagger = \mathbf{H}^*, \quad (68)$$

where  $\mathbf{H}^*$  is the conjugate transpose of  $\mathbf{H}$  and is found by taking the transpose of  $\mathbf{H}$  followed by the complex conjugate of each element in  $\mathbf{H}$ . Since the compression matrix is composed of elements containing just 0 or 1 the pseudoinverse is just the transpose  $\mathbf{H}^\dagger = \mathbf{H}^T$ .

Finally, in (67) we can replace the pseudoinverse of  $\mathbf{H}$  with the transpose; thus the reconstruction process becomes

$$\hat{\mathbf{c}} = \mathbf{H}^T \mathbf{y}. \quad (69)$$

We can improve the performance even more if we assume that the compression is performed as discussed above, that is, where  $m$  coefficients with the largest magnitudes are selected. In that case then  $\mathbf{H}$  is full of zeros except for elements that correspond to the coefficients with largest magnitudes. Using this information we simplify  $l_2$  reconstruction algorithm to just generate a vector of zeros and then copy the stored coefficients into their proper locations in the reconstructed vector. This replaces the matrix-vector multiplication with  $m$  value copies. The updated form of  $l_2$  reconstruction algorithm that does not require the potentially expensive calculation of the matrix pseudoinverse or matrix-vector multiplication is presented in Algorithm 8.

To examine the performance benefit of the new  $l_2$  reconstruction algorithm a compressed form of the occupancy grid seen in Figure 11 is reconstructed using the original algorithm explained in Section 4.1 and the new algorithm described in Algorithm 8. The algorithm is run for  $x$  varying compressed data sizes in  $[0\%, 100\%]$  and the timing results are seen in Figure 11. As seen from the results the run time for the new  $l_2$  reconstruction remains small as the size of the compressed signal grows while the original  $l_2$  reconstruction approach slows down as the size of the compressed signal grows.

## 6. Experimental Results

We would now like to examine how using compressed occupancy grids affects the FastSLAM OG algorithm. Based on the timing results presented in Section 4 only two of the uncompression methods,  $l_2$  (Section 4.1) and CS (Section 4.2), were

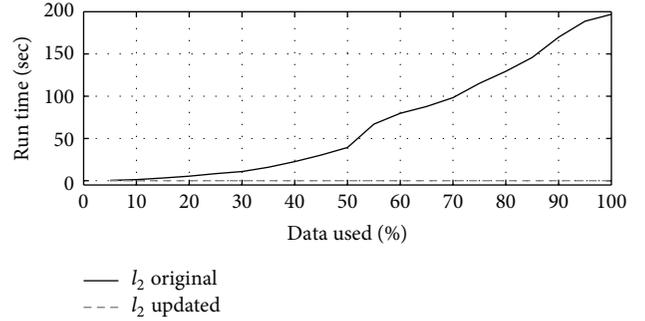


FIGURE 11: The comparison in run time for the original  $l_2$  reconstruction along with the updated form of the reconstruction algorithm.

selected to use in the FastSLAM COG algorithm. Both of the algorithms are compared using data captured by a small unmanned ground vehicle (UGV). The UGV is equipped with Mecanum wheels [28] which allow the vehicle to move in any direction without the constraints that are typical with many ground vehicles. The use of Mecanum wheels makes it difficult to use standard wheel encoders for localization purposes; to overcome this a downward facing camera is attached to the UGV which provides visual odometry data for localization using the approach described in [29]. Other sensors attached to the UGV include a digital compass to provide the global heading of the vehicle and a Hokuyo UTM-30LX LiDAR sensor, with reported accuracy of  $0.1$  to  $10 \text{ m} \pm 30 \text{ mm}$  and  $10$  to  $30 \text{ m} \pm 50 \text{ mm}$ , to provide range and bearing measurements to objects in the environment. In order to provide a “ground truth” trajectory for our test, a Hagisonic StarGazer indoor localization system was used. This sensor uses an upward facing camera and static landmarks placed on the ceiling to triangulate a robotic vehicle in an environment. This sensor has a precision between  $1 \text{ mm}$  and  $10 \text{ mm}$  based on how many landmarks the sensor can see at a given time. This accuracy was much higher than the visual odometry algorithm that provided the odometry input to the SLAM algorithm; thus this trajectory was used to compare the effect of using compressed occupancy grids in our FastSLAM algorithm as opposed to the full occupancy grid.

For comparison purposes, the standard FastSLAM OG algorithm described in Algorithm 1 was used to generate a baseline vehicle path and occupancy grid map. Once the baseline was generated the FastSLAM COG algorithm described in Algorithm 3 was run using the same data used by the standard approach. For each of the two reconstruction algorithms the compression method used is the optimized approach described in Section 5.1 and  $l_2$  approach is implemented as described in Section 5.2. For the CS reconstruction algorithm the convex optimization problem is solved using the NESTA library.

The UGV was remotely driven around a small enclosed  $10 \text{ ft} \times 12 \text{ ft}$  environment while all of the sensor data was logged to memory on a small attached computer. The data was then postprocessed to produce a baseline vehicle path and an occupancy grid representing the testing environment. The baseline vehicle path, along with the true vehicle path, can be

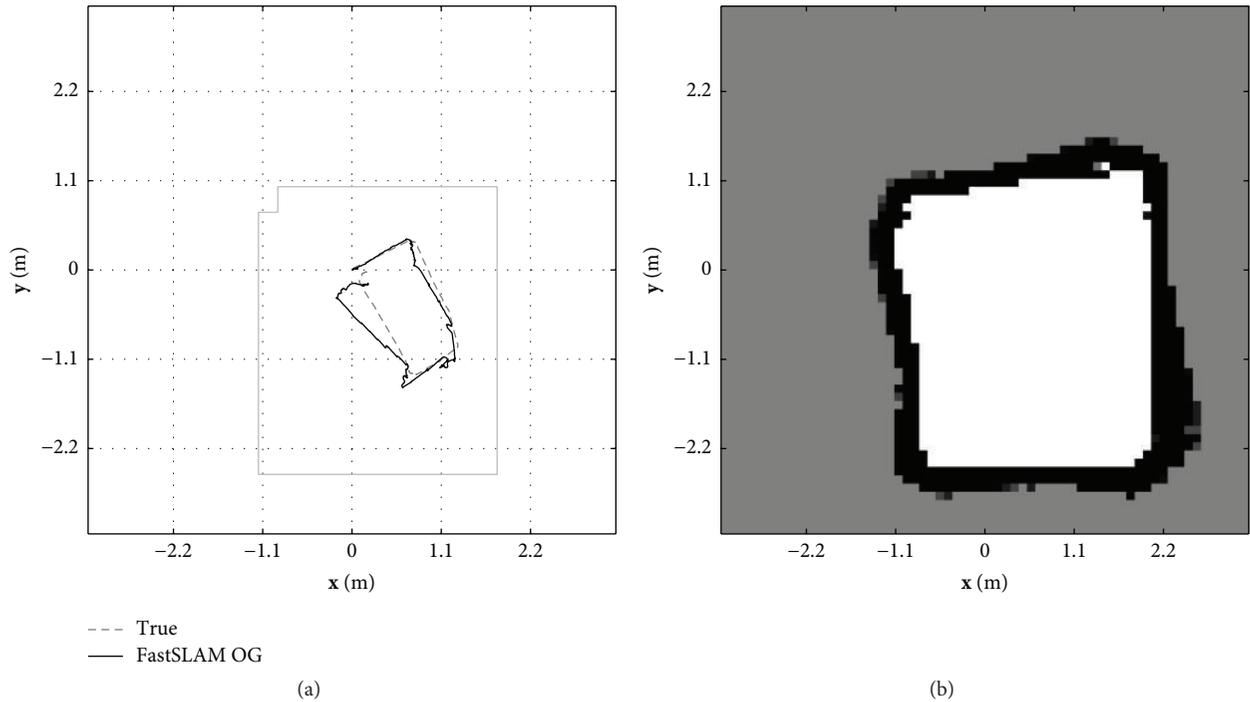


FIGURE 12: The baseline path generated by the FastSLAM algorithm with occupancy grids algorithm along with the true path of the vehicle (a) and the occupancy grid generated by the algorithm that represents the environment in which the vehicle is operating (b).

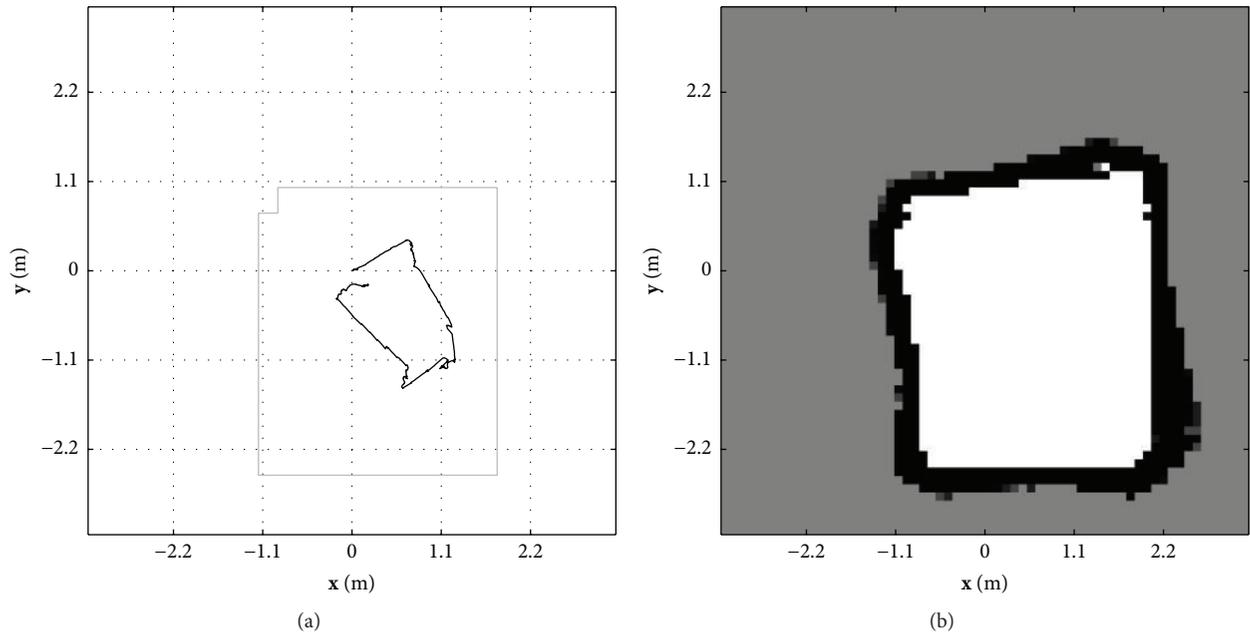


FIGURE 13: The path generated by the FastSLAM COG algorithm along with  $l_2$  reconstruction method using 40% of the original data (a) and the reconstructed occupancy grid generated by the algorithm (b).

seen in Figure 12(a) and the occupancy grid with a 0.1 m cell size is shown in Figure 12(b). The environment used to generate the baseline shown in Figures 12(a) and 12(b) is a simple rectangular indoor environment. Our choice of this simple environment, as opposed to a more complex environment, was made for two reasons. First, in order to properly compare

how the use of compressed occupancy grids affected the localization and mapping an accurate baseline was required for comparison purposes. Based on the sensors that we had at our disposal, an indoor testing was more ideally suited for this task. Secondly, the goal of this research was not to examine how well an occupancy grid based SLAM algorithm

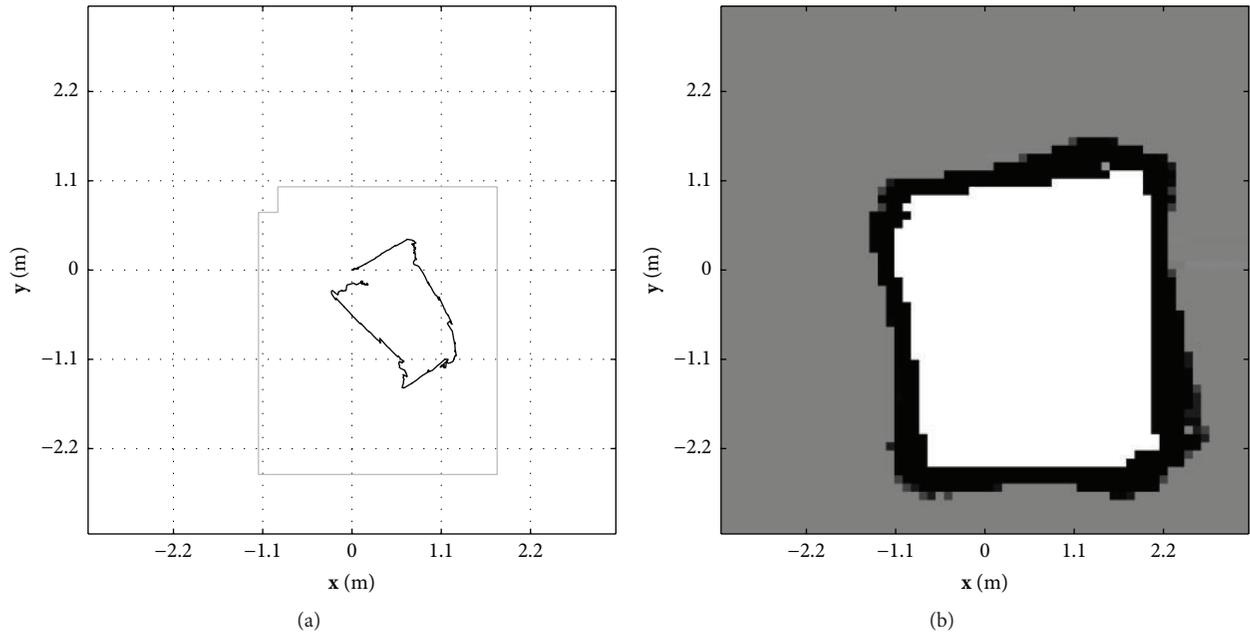


FIGURE 14: The path generated by the FastSLAM COG algorithm using 40% of the original data (a) and the reconstructed occupancy grid generated by the algorithm (b).

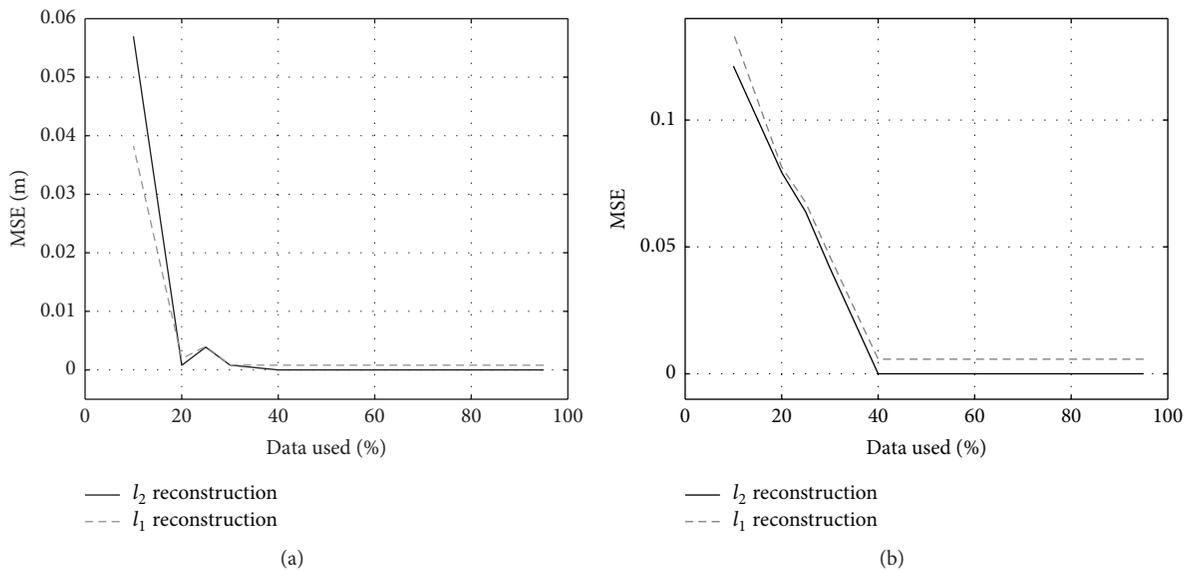


FIGURE 15: The MSE between the path generated by the original algorithm and that generated by the compressed version as a function of the amount of data stored (a) and the MSE between the occupancy grid generated by the original algorithm and that generated using the compressed form of the algorithm as a function of the amount of data stored (b).

performs in complex environment that was beyond our scope and addressed by others; rather our goal was to examine how the performance of the approach was affected by compressing the occupancy grid used to represent the environment, thus testing in complex environment was not attempted for this initial research.

The vehicle path estimate generated by FastSLAM COG algorithm using  $l_2$  reconstruction can be seen in Figure 13(a) and the reconstructed occupancy grid is shown in

Figure 13(b). The generated path estimate using compressed sensing reconstruction approach is shown in Figure 14(a) and the reconstructed occupancy grid is shown in Figure 14(b). For each of the algorithms 40% of the amount of data used to represent the full occupancy grid is used. Figure 15 shows the mean square error (MSE) in the path estimate and reconstructed occupancy grid between the uncompressed form of the algorithm and the compressed form of the algorithm. As Figure 15 shows using more than 40% of

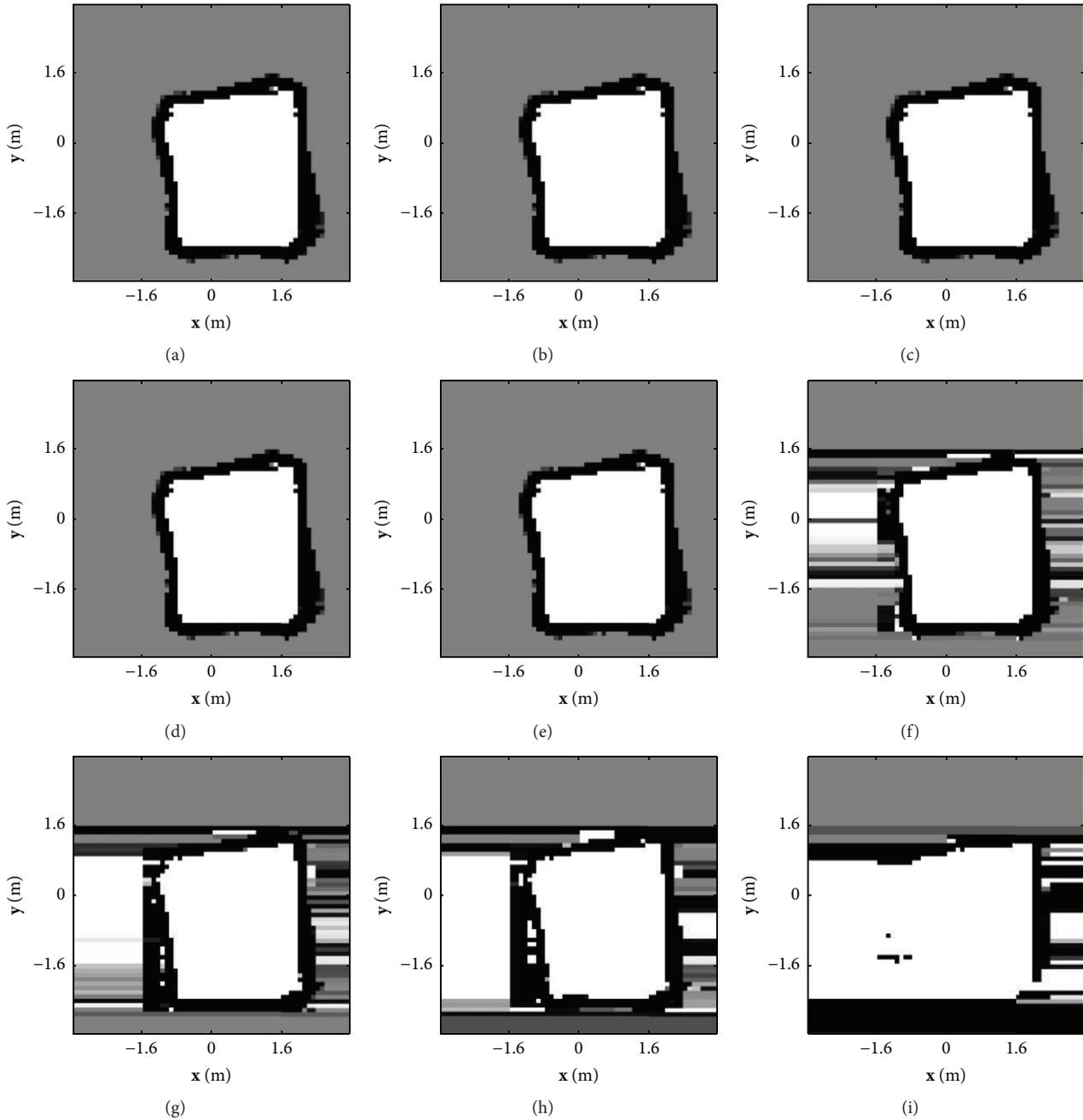


FIGURE 16: A series of final occupancy grids generated by the FastSLAM COG algorithm using  $l_2$  reconstruction approach using 75% (a), 70% (b), 60% (c), 50% (d), 40% (e), 30% (f), 25% (g), 20% (h), and 10% (i) of the full amount of data.

the data does not decrease the MSE any more than the error that exists when using 40% of the data.

The plots provided in Figure 16 illustrate how a decrease in the amount of stored data decreases the quality of the reconstructed occupancy grid. This set of plots shows a series of occupancy grids that were generated as the final map of by the FastSLAM COG algorithm using  $l_2$  reconstruction approach for decreasing amounts of data. As can be seen from the first several grids, up to 40% of the data show no decrease in quality; however as the amount of data used continues

to decrease the amount of error in the final occupancy grid increases until the grid becomes unusable. A similar set of plots are provided in Figure 17 for the FastSLAM COG algorithm using the compressed sensing reconstruction approach and similar results can be seen.

By examining Figure 15 it can be seen that there is a constant error that appears in both the path estimate and the occupancy grid reconstruction from using the compressed sensing approach. This error comes from the parameters that are chosen for the NESTA library. Of most significance

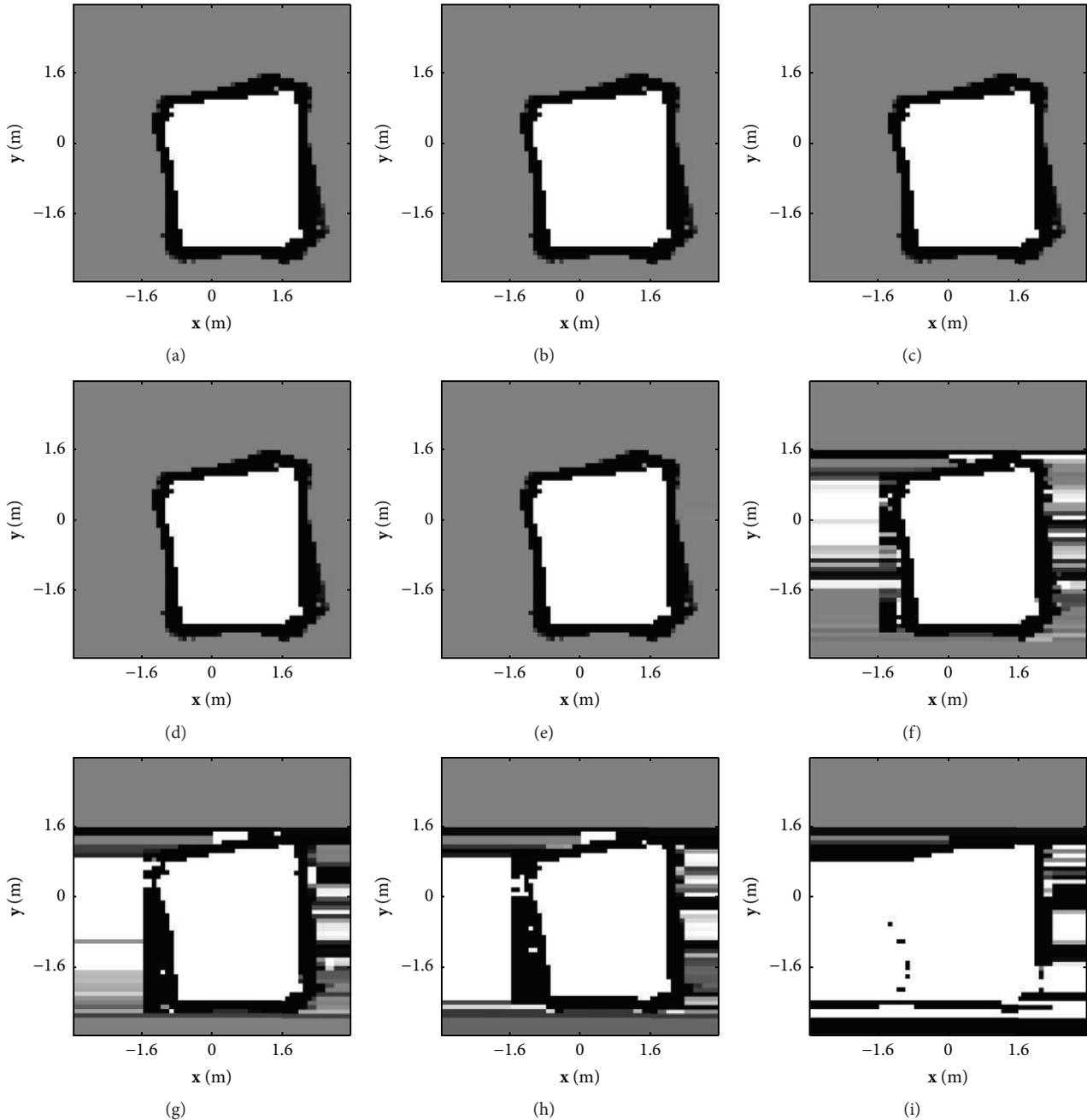


FIGURE 17: A series of final occupancy grids generated by the FastSLAM COG algorithm using the compressed sensing reconstruction approach using 75% (a), 70% (b), 60% (c), 50% (d), 40% (e), 30% (f), 25% (g), 20% (h), and 10% (i) of the full amount of data.

is the value of  $\varepsilon$  from (36) which is the amount of error that is allowed between the coefficients that represent the compressed form of the occupancy grid and the value of those coefficients in the reconstructed occupancy grid. For the presented results the value of  $\varepsilon$  was chosen to be  $10^{-10}$  which is quite small; however it does allow for small errors to occur in the reconstructed occupancy grid. These errors can be reduced or removed completely, from the estimate by choosing a smaller value of  $\varepsilon$  and adjusting the parameters used by the NESTA library. However, as previously

discussed, the accuracy of the grid reconstruction is not the only property of importance to us when deciding which approach should be used for storing and reconstructing the compressed occupancy grid. The other significant property of each algorithm that must be investigated is the time it takes to execute the algorithm. A comparison of the average time it takes to run a single iteration of the FastSLAM COG algorithm using  $l_2$  approach and the compressed sensing approach for varying amounts of data is shown in Figure 18. As seen in these results the average run time of the FastSLAM

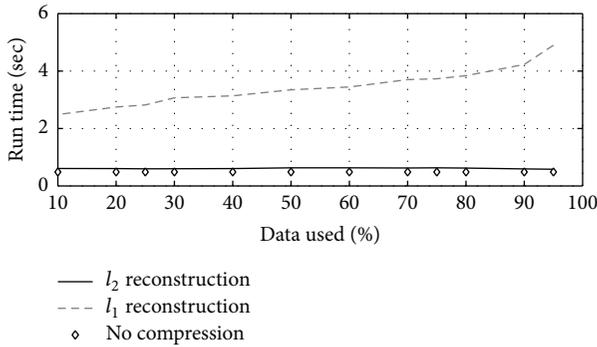


FIGURE 18: The average time to complete a single iteration of the FastSLAM COG algorithm using  $l_2$  reconstruction approach and the compressed sensing reconstruction approach for varying amounts of data stored.

COG algorithm using the compressed sensing approach is significantly longer at each compression level than the algorithm using  $l_2$  reconstruction approach.

**6.1. How Much Compression Can Be Achieved?** There are two key components that affect how much an occupancy grid can be compressed; first is the complexity of the environment in which the UMV is operating. If the environment is simple, such as a rectangular room, then a relatively small amount of data is needed to represent it with respect to a more complex environment. The second component that affects how much an occupancy grid can be compressed is the cell size used. If a large cell size is selected then there are fewer cells to represent the environment compared to when a smaller cell size is selected. It follows that when a larger cell size is selected a larger percentage of the cells must be used to store the information about the environment, even for simple environments so the grid cannot be compressed as much. To illustrate these two components, three separate environments of increasing complexity were generated and a compressed occupancy grid was generated for each environment using four separate cell sizes (1.0 m, 0.5 m, 0.2 m, and 0.1 m) and varying amounts of compression. The first environment is a simple rectangular room and an occupancy grid for this environment generated using the full set of data with the smallest cell size is shown in Figure 19. The MSE as function of data stored for this environment is shown in Figure 20 for each of the four cell sizes. As expected the environment with the largest cell size, when fewer coefficients are needed to store all of the environment information, has the largest error when a small amount of data is used to represent the environment. It can also be seen that, for each of the four cell sizes, as more information is stored, the error in the reconstructed form of the occupancy grid decreases.

A second environment with more complexity is shown in Figure 21 which is a rectangular room with complex corners. The MSE between the true occupancy grid and the reconstructed grid as a function of the amount of data stored is shown in Figure 22 for each of the cell sizes. The overall behavior of the error is the same as in the simple case with the largest cell size having the largest error when small amounts

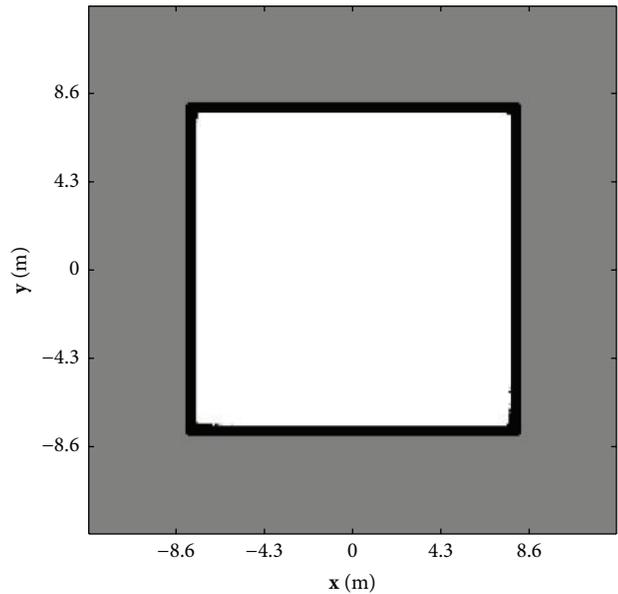


FIGURE 19: Occupancy grid of simple rectangular environment using 100% of the data and a cell size of 0.1 m.

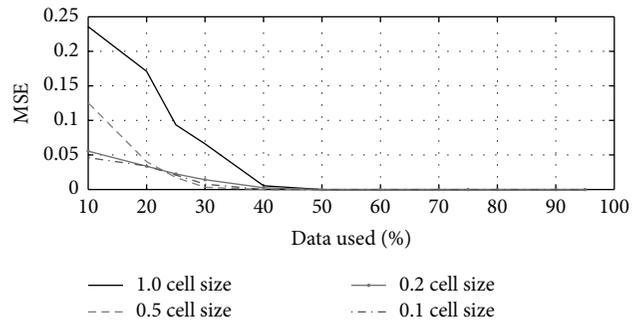


FIGURE 20: MSE between the true occupancy grid of the simple environment and the reconstructed form of the grid as a function of percent of data stored for each of the four cell sizes.

of data are used. However, because of the added complexity in this environment, the errors at the lower data percentages are larger than the simple case which is what we expected because more information is required to store the more complex environment.

Finally, a complex environment of a rectangular room with complex corners along with rectangular objects located throughout the rectangular room is shown in Figure 23. As with the previous two environments the MSE between the true occupancy grid and the reconstructed occupancy as a function of data is shown in Figure 24 for each cell size. As expected the use of larger cell sizes causes more errors in the reconstructed form of the grid when small amounts of data are used. Also, the additional complexity causes the errors at lower data percent to be larger than in the more simple environments. It can be seen from the presented results that there appears to be a “magic” compression level of 40% at which, no matter the complexity, the compressed occupancy can be reconstructed with significant accuracy.

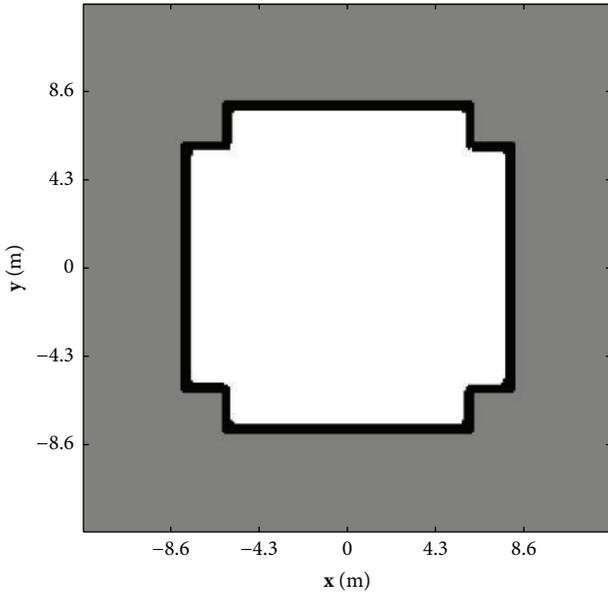


FIGURE 21: Occupancy grid of rectangular environment with complex corners using 100% of the data and a cell size of 0.1 m.

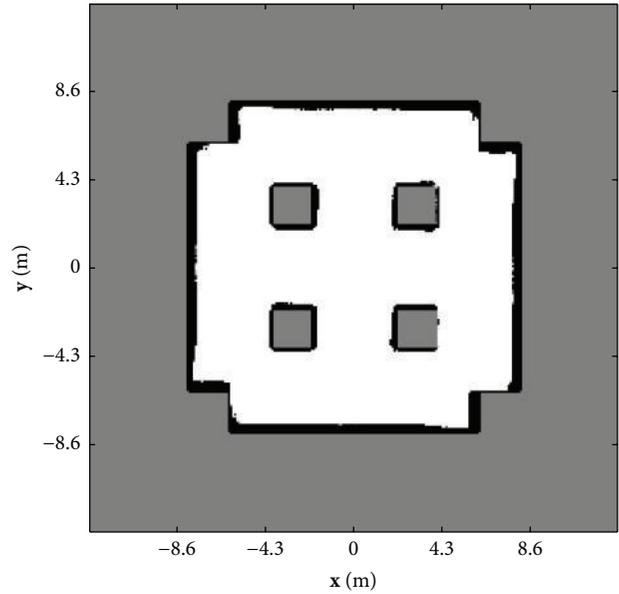


FIGURE 23: Occupancy grid of the complex environment using 100% of the data and a cell size of 0.1 m.

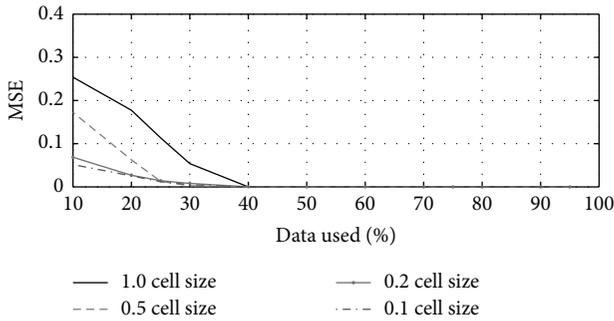


FIGURE 22: MSE between the true occupancy grid of the environment with medium complexity and the reconstructed form of the grid as a function of percent of data stored for each of the four cell sizes.

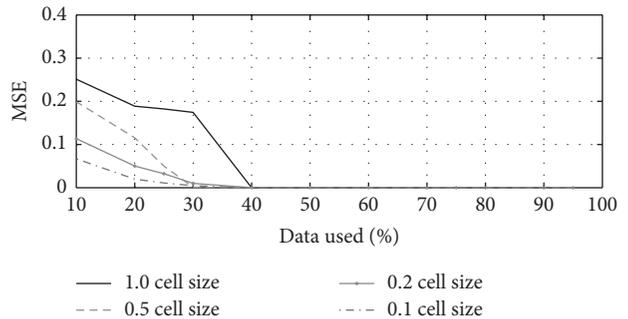


FIGURE 24: MSE between the true occupancy grid of the complex environment and the reconstructed form of the grid as a function of percent of data stored for each of the four cell sizes.

## 7. Conclusions

In this paper an approach for solving the SLAM problem was presented that makes use of compressed occupancy grids. The approach is based on an extension to the FastSLAM algorithm that represents an unknown environment by using occupancy grids. A modified form of the SLAM algorithm, FastSLAM COG, was presented in Section 3 which makes use of a generic compression and reconstruction method to solve the SLAM problem. Four specific reconstruction methods were examined in Section 4 in order to see how well each of the approaches could reconstruct a compressed signal along with an analysis of the time needed for each algorithm to complete. Based on the reconstruction time, two of the four algorithms were used to replace the generic reconstruction method in the FastSLAM COG algorithm and their performance was examined in Section 6 using experimental data. From the presented experimental results

it was concluded that the FastSLAM COG algorithm using  $l_2$  reconstruction approach could solve the SLAM problem with almost no errors compared to the FastSLAM OG algorithm while storing only 40% of the data required to store the complete occupancy grid. The experimental results presented in this paper were achieved by exploring how compressing the occupancy grid affects the performance of the algorithm in simple environments. In order to further examine our approach, a next step would be to examine how well this approach performs in more complex and unstructured environments. Also, further investigation into the effect of using more complex compression and reconstruction approaches as well as the possibility of dynamically resizing the cell size could lead to higher data compression rates without inducing errors into the SLAM solution.

## Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

## Acknowledgments

This material is based upon work supported by (while serving at) the National Science Foundation of the United States.

## References

- [1] A. Elfes, *Occupancy grids: a probabilistic framework for robot perception and navigation* [Ph.D. thesis], Carnegie Mellon University, Department of Electrical and Computer Engineering, Pittsburgh, Pa, USA, 1989.
- [2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fast-SLAM: a factored solution to the simultaneous localization and mapping problem," in *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI '02) and the 14th Innovative Applications of Artificial Intelligence Conference (IAAI '02)*, pp. 593–598, Edmonton, Canada, August 2002.
- [3] C. Radhakrishna Rao, "Information and accuracy attainable in the estimation of statistical parameters," *Bulletin of the Calcutta Mathematical Society*, vol. 37, no. 3, pp. 81–91, 1945.
- [4] D. Blackwell, "Conditional expectation and unbiased sequential estimation," *The Annals of Mathematical Statistics*, vol. 18, pp. 105–110, 1947.
- [5] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian Bayesian state estimation," *IEE Proceedings Part F: Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.
- [6] A. F. Smith and A. E. Gelfand, "Bayesian statistics without tears: a sampling-resampling perspective," *The American Statistician*, vol. 46, no. 2, pp. 84–88, 1992.
- [7] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.
- [8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT press, Cambridge, Mass, USA, 2006.
- [9] R. Baraniuk, M. A. Davenport, M. F. Duarte, and C. Hegde, "An introduction to compressive sensing," *Connexions E-Textbook*, 2011.
- [10] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. 23, no. 1, pp. 90–93, 1974.
- [11] E. Moore, "On the reciprocal of the general algebraic matrix," *Bulletin of the American Mathematical Society*, no. 26, pp. 294–295, 1920.
- [12] R. Penrose, "A generalized inverse for matrices," in *Mathematical proceedings of the Cambridge philosophical society*, vol. 51, no. 3, pp. 406–413, Cambridge University Press, 1955.
- [13] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [14] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B: Methodological*, vol. 58, no. 1, pp. 267–288, 1996.
- [15] S. Becker, J. Bobin, and E. J. Candès, "Nesta: a fast and accurate first-order method for sparse recovery," *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 1–39, 2011.
- [16] E. Candès and J. Romberg, " $\ell_1$ -Magic: Recovery of Sparse Signals via Convex Programming," vol. 4, 2005, <http://users.ece.gatech.edu/justin/llmagic/downloads/llmagic.pdf>.
- [17] A. Carmi, P. Gurfil, and D. Kanevsky, "Methods for sparse signal recovery using Kalman filtering with embedded pseudo-measurement norms and quasi-norms," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2405–2409, 2010.
- [18] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [19] S. J. Julier and J. LaViola Jr., "On Kalman filtering with nonlinear equality constraints," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2774–2784, 2007.
- [20] R. Chartrand, "Exact reconstruction of sparse signals via non-convex minimization," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 707–710, 2007.
- [21] R. G. Brown and P. Hwang, *Introduction to Random Signals and Applied Kalman Filtering: With MATLAB Exercises and Solutions*, John Wiley & Sons, New York, NY, USA, 3rd edition, 1997.
- [22] T. H. Cormen, *Introduction to Algorithms*, McGraw-Hill Higher Education, 2001.
- [23] G. Arfken, *Mathematical Methods for Physicists*, Academic Press, Orlando, Fla, USA, 3rd edition, 1985.
- [24] M. H. Lee and M. Kaveh, "Fast hadamard transform based on a simple matrix factorization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 6, pp. 1666–1667, 1986.
- [25] R. Coifman, F. Geshwind, and Y. Meyer, "Noiselets," *Applied and Computational Harmonic Analysis*, vol. 10, no. 1, pp. 27–44, 2001.
- [26] MathWorks, "Moore-penrose pseudoinverse of matrix," 2014, <http://www.mathworks.com/help/matlab/ref/pinv.html>.
- [27] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, vol. 50, Society for Industrial and Applied Mathematics, 1997.
- [28] O. Diegel, A. Badve, G. Bright, J. Potgieter, and S. Tlale, "Improved mecanum wheel design for omni-directional robots," in *Proceedings of the 2002 Australasian Conference on Robotics and Automation*, pp. 117–121, Auckland, New Zealand, 2002.
- [29] C. H. Cain and A. Leonessa, "Testing vision-based sensors for enclosed underwater environments when applied to ekf slam," in *Proceedings of the 5th Annual Dynamic Systems (ASME '12) and Control Conference Joint with the 11th Motion and Vibration Conference (JSME '12)*, pp. 213–220, American Society of Mechanical Engineers, 2012.

## Research Article

# A New Technique for Integrating MEMS-Based Low-Cost IMU and GPS in Vehicular Navigation

Neda Navidi,<sup>1</sup> René Jr. Landry,<sup>1</sup> Jianhua Cheng,<sup>2</sup> and Denis Gingras<sup>3</sup>

<sup>1</sup>LASSEN Laboratory, École de Technologie Supérieure, 1100 Notre-Dame Street West, Montreal, QC, Canada H3C 1K3

<sup>2</sup>Marine Navigation Research Institute, College of Automation, Harbin Engineering University, Harbin 150001, China

<sup>3</sup>LIV Laboratory, Electrical and Computer Engineering Department, Université de Sherbrooke, Sherbrooke, QC, Canada J1K 2R1

Correspondence should be addressed to Neda Navidi; [neda.nav3@gmail.com](mailto:neda.nav3@gmail.com)

Received 20 November 2015; Revised 4 April 2016; Accepted 28 April 2016

Academic Editor: Maan E. El Najjar

Copyright © 2016 Neda Navidi 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.

In providing acceptable navigational solutions, Location-Based Services (LBS) in land navigation rely mostly on integration of Global Positioning System (GPS) and Inertial Navigation System (INS) measurements for accuracy and robustness. The GPS/INS integrated system can provide better land-navigation solutions than the ones any standalone system can provide. Low-cost Inertial Measurement Units (IMUs), based on Microelectromechanical Systems (MEMS) technology, revolutionized the land-navigation system by virtue of their low-cost miniaturization and widespread availability. However, their accuracy is strongly affected by their inherent systematic and stochastic errors, which depend mainly on environmental conditions. The environmental noise and nonlinearities prevent obtaining optimal localization estimates in Land Vehicular Navigation (LVN) while using traditional Kalman Filters (KF). The main goal of this paper is to effectively eliminate stochastic errors of MEMS-based IMUs. The proposed solution is divided into two main components: (1) improving noise cancellation, using advanced stochastic error models in MEMS-based IMUs based on combined Autoregressive Processes (ARP) and first-order Gauss-Markov Process (1GMP), and (2) modeling the low-cost GPS/INS integration, using a hybrid Fuzzy Inference System (FIS) and Second-Order Extended Kalman Filter (SOEKF). The results obtained show that the proposed methods perform better than the traditional techniques do in different stochastic and dynamic situations.

## 1. Introduction

Steadily increasing economical and environmental constraints as well as stringent safety requirements have triggered the development and widespread use of low-cost Land Vehicular Navigation (LVN) systems over the last decade. LVNs have many applications: nonsafety applications, such as fleet management and traffic optimization, or active safety applications, such as Collision Mitigation Braking Systems (CMBS) and lane keeping systems. LVNs can also be used for protecting vehicles from theft in vehicle tracking systems or for reducing the greenhouse gas emissions in environmental monitoring systems. Besides, they can also be used in autonomous car navigation systems and emergency assistance services. Furthermore, the need for recognizing driver's behavior in many applications requires low-cost and reliable

estimation of vehicular position and attitude. Clearly, universal use of LVN in road transportation demands lower cost, widespread availability, and ever improving performance [1].

Advanced commercial Location-Based Services (LBS) must be able to solve difficult positioning problems, especially those in indoor-parking, urban canyons, or dense foliage situations, by providing acceptable customer support, whenever the GPS signal is lost. One issue that is recently investigated in positioning-based systems relies on multi-Global Navigation Satellite Systems (GNSS) environment which is realized due to the effective usage of the global constellation of GLONASS, satellites of Galileo, and Chinese BeiDou Navigation Satellite System (BDS). Multi-GNSS environment offers several advantages over a standalone GPS. Indeed, multi-GNSS environment may increase the availability of navigation satellites where the received GPS signals are less

than 4 satellites in urban canyons. Moreover, it can improve the accuracy of positioning, especially when the navigation system considers the horizontal positioning accuracy [2].

However, multi-GNSS-based techniques may increase the intersystem interference, particularly when multi-GNSS environments broadcast navigation signals in overlapped frequency bands. Moreover, these techniques impose higher levels of complexity, because the analog-part of the receiver should deal with multiple-system, multiple frequencies and considerable bandwidths [3]. Finally, the mentioned developed techniques consist in postprocessing techniques that require laborious manipulations and calculations, which may not be suitable for real-time commercial applications. A modern approach to solve the current positioning problems is to integrate the GPS signals with Low-cost Inertial Measurement Units (IMUs) data. IMUs can estimate the position and the attitude of the vehicle by employing an Inertial Navigation System (INS) mechanization process (accelerometers and gyroscopes), which is an integration of linear accelerations and angular rates. However, MEMS-based low-cost IMUs perform accurately just for a limited time as their positioning and attitudinal errors increase steadily with time.

MEMS-based IMUs are classified based on their mass position recognition, their operation mode, and their fabrication procedure. Piezoresistive, Capacitive, Piezoelectric, and Resonant element sensors are different types of MEMS sensors which are placed in the classification of mass position recognition [4]. These types of sensors transform the physical work caused by variation of movement into an electrical signal. Piezoresistive and Piezoelectric sensors are widely used in MEMS-based IMUs because of their simplicity, cost-effectiveness, and wide dynamic range. However, they suffer from low precision and high sensitivity to temperature [4]. Performance characteristics of MEMS-based sensors include bandwidth, noise floor, misalignment, drift, linearity, dynamic range, and power consumption. Thus, the MEMS-based inertial sensor errors are generally composed of stochastic errors which have an undesirable impact on the quality of the navigation solution [5].

The first part of the proposed methodology consists of augmenting the navigation solution with 1st-order Gauss-Markov Process (IGMP) for dynamically estimating and compensating the stochastic errors of MEMS-based low-cost IMUs. Two different identification methods were exploited in this study to identify the IGMP parameters. The first one is a traditional method based on the analysis of Autocorrelation Function (ACF) and the second one is based on analysis of 1st-order Autoregressive Process (ARP) model. ACF does not fully match the random error characteristics of low-cost MEMS-based IMUs. What mainly causes this mismatch is the high nonlinearity of the low-cost inertial sensor errors leading to higher order GMPs (i.e., 2nd- or 3rd-order GMPs) distributions instead of IGMP [6]. Thus, there is a need for the development of better model-based methods to accurately estimate the distribution of the IGMP low-cost inertial sensor errors. This study proposes using combined IGMP-ARP method as an alternative candidate to identify the IGMP's parameters distribution in the low-cost inertial sensors. To estimate the ARP's coefficients,

three different algorithms were considered and implemented, namely, the Yule-Walker (YW) method, the Burg method, and the modified-covariance method [7].

As these static errors have both high frequency (HF) term and low frequency (LF) term, decreasing both of them is needed in order to develop the accuracy of the low-cost inertial sensors. Wavelet Denoising Techniques (WDTs) have been utilized in similar research studies, due to their significant role in removing the HF noises [8]. So, it is presented a mixture of the WDT (with different levels) and IGMP to evaluate the accuracy improvement of the low-cost inertial sensors when these methods are blended together.

Traditionally, GPS/INS integration is carried out by using Kalman Filter (KF) techniques. This approach has been validated in many studies, where high-end IMUs were used [9]. However, recent studies report several shortcomings in KF-based INS/GPS integration, when using low-cost MEMS-based inertial sensors [6]. Dynamic models that define the navigation problem are originally nonlinear [10], but, to allow the use of simple linear filters such as the KF, these models are generally linearized under a small error assumption. With the assumption made by the KF that the system is characterized by a linear model, driven by white Gaussian noise, it is not possible to describe the nonlinear error dynamics of MEMS-based INS [11].

The conventional 1st-order EKF is used extensively for integration of high-end INS and GPS data, which can achieve a suboptimal estimation of Minimum Mean Square Error (MMSE) of the vehicle state vector [12]. 1st-order Extended Kalman Filter (EKF) is based on the assumption that the higher order terms (2nd-order and up) of a Taylor series expansion are small enough to be ignored. This condition may hold for high-end navigation systems where the errors are kept small and bounded, but not for highly nonlinear systems, such as MEMS-based INS. In addition, to provide an acceptable position and attitude estimates, the 1st-order EKF requires proper nonlinear models of vehicle dynamics and measurement sensors. The first-order linearization, especially in the presence of highly nonlinear dynamics, often leads to divergence of the state covariance matrix [11]. Thus, using such assumption for a low-cost navigation system may result in unstable solution, as demonstrated by several previously published results [13].

Several research groups investigated the possibility of using sampling techniques to cope with highly non-Gaussian multimodal error distributions in GPS/INS integration and thus avoiding linearization of the dynamic models [13]. Unscented Kalman Filter (UKF) and Particle Filter (PF) are two such alternative methods that can provide better performance than does classical KF or EKF, but at high computational costs [11, 14]. Some authors assert that the limitations of the sampling and the Kalman-based techniques can be overcome by using Artificial Intelligence (AI) [15]. The integration of AI algorithms with next generation embedded navigation systems is made possible with the technological advancement in computer technologies. In the last decade, various neural networks (NN) and Fuzzy Inference Systems- (FIS-) based integration scheme were introduced [16–18]. AI algorithms require a training phase, which is realized

during GPS signal availability using a learning algorithm. However, AI-based GPS/INS integration schemes suffer from two main drawbacks: these methods generally require long training time that greatly limits their utilization in a real-time commercial application; moreover, they highly depend on the consistency of the training data and show limited success if the error dynamics do not stay the same between their training and prediction stages [11].

The second part of the proposed approach is to combine both linearized navigation filters and AI techniques into an intelligent cognitive navigator. SOEKF as the linearized navigation filter helps in reducing the linearization error during the filtering process and FIS-based approach as an AI technique, predicting the error states of the SOEKF based on a Covariance Matching Estimation Technique (COMET). In fact, the dynamic characteristics of vehicle motion in body frame and the navigation frame form based on the SOEKF process. The FIS can be exploited to increase the accuracy and the robustness of the SOEKF and to prevent its divergence in the tuning phase of SOEKF. The performances of the proposed methods were evaluated by a road test, using Matlab. The results show that the proposed method is effective in reducing the root mean square error (RMSE) of position by 45% compared with the SOEKF.

The remainder of the paper is organized as follows: Section 2 describes the main methods of this study. Section 3 deals with the proposed methods and the contributions of this paper for GPS/INS integration and the noise modeling. The method employed here for the GPS/INS integration is based on the proposed hybrid FIS-SOEKF model. Additionally, the proposed noise modeling of low-cost inertial sensor utilizes the combined IGMP-ARP. Section 4 deals with the road test and the detailed analysis of experimental data, followed by a discussion on the performance obtained. Finally, Section 5 concludes this paper by summarizing the results achieved and by offering suggestions for future work to improve the proposed methods.

## 2. Background

This section presents details on background of techniques and ideas used in this study. More specifically, two main topics will be studied in this section: Second-Order Extended Kalman Filter (SOEKF) as a data-fusion technique for automotive navigation and stochastic error modeling technique in Inertial Navigation Systems.

**2.1. Second-Order Extended Kalman Filter (SOEKF).** Linear state estimators have been widely used in the early literature for guidance, navigation, and control of various types of vehicles, including aircraft, space shuttles, ships, submarines, and automotive land vehicles [9]. As mentioned in Introduction, in order to perform optimal estimation using the classical Kalman Filter-based technique on the nonlinear and stochastic INS dynamic model, the model must be linearized around assumptions. First-order and second-order Extended Kalman Filter (EKF) are two Kalman Filter-based techniques which linearized based on the assumption of first and second orders of a Taylor series expansion.

In this study, SOEKF was addressed for proposed data-fusion of low-cost INS and GPS aided with FIS. Thus, a summary of SOEKF performance is presented in this section. The SOEKF is an estimator method, which can solve the problem of estimating the state of a controlled process. The model of SOEKF is

$$\begin{aligned}\bar{\bar{x}}_k &= f(x_{k-1}, k-1) + q_{k-1}, \\ \bar{\bar{y}}_k &= h(x_k, k) + r_k,\end{aligned}\quad (1)$$

where  $\bar{\bar{x}}_k \in \mathbb{R}^m$  is the state,  $\bar{\bar{y}}_k \in \mathbb{R}^m$  is the measurement,  $q_{k-1} \sim N(0, Q_k)$  is the process noise,  $r_k \sim N(0, R_k)$  is measurement noise, and  $f$  and  $h$  are the dynamic and measurement model functions. The prediction and correction steps of SOEKF are described by [19–21]

$$\begin{aligned}\hat{x}_k^- &= f(x_{k-1}, k-1) + \frac{1}{2} \sum_i e_i \text{tr} \{F_{xx}^{(i)}(x_{k-1}, k-1) \\ &\cdot P_{k-1}\}, \\ P_k^- &= F_x(x_{k-1}, k-1) P_{k-1} F_x^T(\hat{x}_{k-1}, k-1) + \frac{1}{2} \sum_i e_i \\ &\cdot e_i^T \text{tr} \{F_{xx}^{(i)}(x_{k-1}, k-1) P_{k-1} F_{xx}^{(i')}(\hat{x}_{k-1}, k-1) P_{k-1}\} \\ &+ \eta_k,\end{aligned}\quad (2)$$

where  $\hat{x}_k^-$  and  $P_k^-$  are, respectively, the time-propagated state estimates and covariance  $F_k$  and  $H_k$  are the state transition and the design matrices;  $x_k$  is the error state vector and  $z_k$  the measurement vector:

$$v_k = z_k - h(x_k^-, k) + \frac{1}{2} \sum_i e_i \text{tr} \{H_{xx}^{(i)}(x_k, k) P_k\}, \quad (3)$$

$$\begin{aligned}\text{COV}_k &= H_x(x_k^-, k) P_k^- H_x^T(x_k^-, k) \\ &+ \frac{1}{2} \sum_{i,i'} e_i e_{i'}^T \text{tr} \{H_{xx}^{(i)}(x_k^-, k) P_k^- H_{xx}^{(i')}(\hat{x}_k^-, k) P_k\} \\ &+ \mu_k,\end{aligned}\quad (4)$$

$$K_k = P_k^- H_x^T(x_k^-, k) + \text{COV}_k^{-1}, \quad (5)$$

$$x_k = x_k^- + K_k v_k, \quad (6)$$

$$P_k = P_k^- - K_k S_k K_k^T, \quad (7)$$

where  $F_x(x, k-1)$  and  $H_x(x, k)$  are Jacobian matrices of  $f_i$  and  $h_i$  which are expressed by

$$\begin{aligned}[F_x(x, k-1)]_{i,i'} &= \frac{\partial f_i(x, k-1)}{\partial x_{i'}}, \\ [H_x(x, k)]_{i,i'} &= \frac{\partial h_i(x, k)}{\partial x_{i'}},\end{aligned}\quad (8)$$

$F_{xx}^{(i)}(x, k-1)$  and  $H_{xx}^{(i)}(x, k)$  are the Hessian matrices of  $f_i$  and  $h_i$  which are defined by

$$\begin{aligned} [F_{xx}^{(i)}(x, k-1)]_{i,i'} &= \frac{\partial^2 f_i(x, k-1)}{\partial x_i \partial x_{i'}}, \\ [H_{xx}^{(i)}(x, k)]_{i,i'} &= \frac{\partial^2 h_i(x, k)}{\partial x_i \partial x_{i'}}, \end{aligned} \quad (9)$$

and  $e_i$  is the unit vector in direction of the coordinate axis  $i$ .

**2.2. Stochastic Error Modeling Technique in Inertial Navigation Systems.** Techniques developed for the correction of inertial sensor errors can generally be divided into two categories that are (1) deterministic error calibration procedures and (2) stochastic error estimation models. Only the stochastic error estimation models are considered in this work.

In general, stochastic errors can be separated into two categories, that is, white noise and colored (or correlated) noise. White noise is a random process characterized by an Autocorrelation Function (ACF) of the form of a Dirac. On the other hand, colored noise may exhibit different ACFs depending on the nature of the noise (pink, red, grey, etc.).

White noise can generally be mitigated using adequate low-pass filter or Wavelet Denoising Technique (WDT) while colored noise must be precisely modeled in order to be estimated and compensated. Traditionally, colored noise of IMUs is modeled as a first-order Gauss-Markov Process (IGMP) with an exponentially decaying ACF. This section presents details on stochastic processes to model the colored noise as well as Wavelet Denoising Technique (WDT) to eliminate the white noise in IMUs.

**2.2.1. Stochastic Processes.** The most important errors to consider are stochastic errors since these errors, if not dealt with, will have a significant negative impact on the accuracy of vehicular tracking output [5]. Several research groups started investigating the possibility of using Mont-Carlo simulation technique for modeling the stochastic errors of MEMS devices due to multicomponent character of the sensors. However, it increases the number of degrees of freedom in the dynamic model [22]. So the alternative methods like generalized stochastic perturbation technique, Stochastic Finite Element Method (SFEM), and Second-Order Second Moment (SOSM) approach are considered in [23–25].

Although there are several stochastic processes, this section addresses only those that were considered for this study, namely, first-order Gauss-Markov Process (IGMP) and Autoregressive Processes (ARP). The relation between these two stochastic processes and proposed stochastic error modeling will be detailed in Section 3.1.

**First-Order Gauss-Markov Process (IGMP).** Traditionally, the noises of IMUs are modeled as 1st-order Gauss-Markov Process (IGMP) [8, 26–28]. The continuous-time and the discrete-time models of GMP can be given, respectively, by (10) and (11). Consider

$$\dot{A}(t) = -\frac{A(t)}{T_c} + w(t), \quad (10)$$

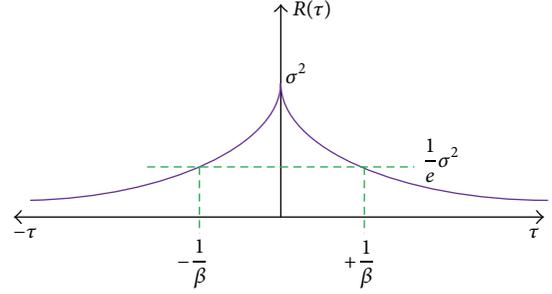


FIGURE 1: Autocorrelation Function (ACF) for IGMP.

$$A_{k+1} = e^{-\Delta t/T_c} \cdot A_k + w(n), \quad (11)$$

where  $A_k$  and  $w(n)$  are the IGMP and the white noise (WN) and  $\Delta t$  and  $T_c$  are the sampling and the correlation times, respectively. The noise covariance can be expressed by

$$\sigma_{wk}^2 = \sigma_{xk}^2 (1 - e^{-2\Delta t/T_c}). \quad (12)$$

Classically, IGMP's parameters can be estimated by Autocorrelation Function (ACF). Several studies investigated the use of ACF in analyzing the stochastic error of the inertial sensors and also to acquire IGMP's parameters. It can be modeled with an exponentially decaying ACF, as shown in Figure 1. The essential parameters required to model this process are  $\sigma^2$  and  $\beta$ , which are given by

$$R(\tau) = \sigma^2 e^{\beta|\tau|}, \quad \beta = \frac{1}{T_c}, \quad (13)$$

where  $\sigma^2$ ,  $\tau$ , and  $\beta$  are the noise covariance, the time lag, and the reciprocal of the process correlation time, respectively. As shown in Figure 1, this process limits the uncertainty of the signal, and this forms the distinctive feature of this process. And, so,  $R(\tau)$ , at any correlation time, is less than or equal to  $R(0)$ . However, the accuracy of ACF results depends on the length of the recorded data.

**Autoregressive Processes (ARP).** ACF might be matched to the higher order of GM, instead of the 1st-order one, for MEMS inertial sensors, as stated by many workers [29]. Thus, the other method to model the errors, which is presented in [29], is Autoregressive Processes (ARP). ARP is one type of Autoregressive Moving Average (ARMA) process, which is produced by the combination of past values and is expressed by

$$A_k = -\sum_{n=1}^b \alpha_n \cdot A_{k-n} + \beta_0 \omega_k, \quad (14)$$

where  $A_k$ ,  $b$ ,  $\alpha_n$ ,  $\beta_0$ , and  $\omega_k$  are the ARP output, order of process, model parameters, standard deviation, and white noise, respectively. Given the value of  $b$ , minimizing the root mean square error (RMSE) between the original signal and the signal estimated by ARP can help us to identify the ARP's coefficients. This error is presented by

$$\epsilon_k = \sum_k \left[ A_k + \sum_{n=1}^b \alpha_n A_{k-n} \right]^2. \quad (15)$$

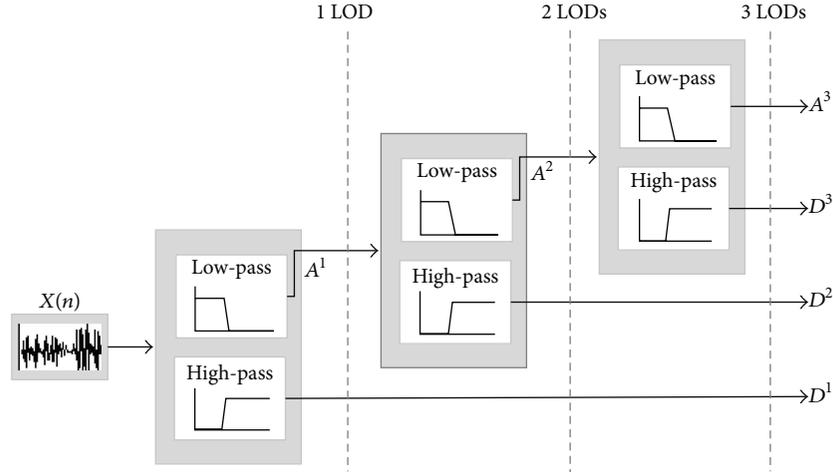


FIGURE 2: Application of Wavelet Denoising Technique (WDT) with different Levels of Decomposition (LODs).

There are several identification algorithms to estimate the ARP coefficients. For this study, three methods, namely, Yule-Walker (YW), Burge's, and modified-covariance (MCOV) algorithms, will be considered for implementation. The method that gives the minimum RMSE will be the best one to estimate the ARP's coefficients. So, an investigation of the best method to estimate the ARP coefficients is considered in this study.

**2.2.2. Wavelet Denoising Technique (WDT).** Wavelet Denoising Technique (WDT) is based on the wavelet filtering method, which can eliminate the stochastic error in the measurements of the inertial sensors at High Frequencies (HFs) without altering important information contained in the signal [30, 31]. A Multiresolution Analysis (MRA) algorithm was performed on the WDT, which can decompose a signal into different subbands with various time and frequency resolutions. Conceptually, in WDT, signal  $x(n)$  is filtered by one high-pass filter and one low-pass filter with downsampling by two ( $\downarrow 2$ ) in the continuous Levels of Decomposition (LODs), as shown in Figure 2.

The maximum amount of information about the original signal  $x(n)$  was contained in the approximation coefficients ( $A^k$ ) in each subband. Hence, the minority information of  $x(n)$  was contained in HF noise components, which were identified by detailed coefficients ( $D^k$ ). This technique can be combined with the different noise modeling process like IGMP and ARP, to investigate the stochastic error of inertial sensors. After applying the technique, the coefficients of IGMP were determined from the enduring noises. More related explanations are presented in Sections 3 and 4.

### 3. Proposed Methods

This paper proposes combining two independent and complementary solutions into a global integrated navigation system to provide stable low-cost ubiquitous automotive navigation in severe urban environments. The first proposed

solution consists of augmenting the navigation solution with combined IGMP-ARP for dynamically estimating and compensating the static errors of low-cost inertial. The second proposed solution is to combine both linearized navigation filters and AI techniques into an intelligent cognitive navigator. SOEKF as the linearized navigation filter helps in reducing the linearization error during the filtering process; and FIS-based approach as an AI technique helps in predicting the error states of the SOEKF. This section presents details on these two proposed solutions.

**3.1. Proposed Stochastic Error Modeling of Low-Cost IMU.** Autocorrelation Function (ACF), which is presented in Section 2, does not fully match the random error characteristics of low-cost MEMS-based IMUs. What mainly causes this mismatch is the high nonlinearity of the low-cost inertial sensor errors leading to other distributions instead of IGMP [6]. Thus, there is a need for the development of better model-based methods to accurately estimate the distribution of the IGMP low-cost inertial sensor errors. This study proposes using a new method, named "combined IGMP-ARP method," as an alternative candidate to identify the IGMP's parameters distribution in the low-cost inertial sensors. The fundamental concept of the model is IGMP. However, its  $\sigma^2$  and  $\beta$  are calculated by 1st-order ARP rather than by ACF. Substituting  $b = 1$  within (14) gives the first-order ARP as follows:

$$A_k = \alpha_1 \cdot A_{k-1} + \beta_0 \omega_k. \quad (16)$$

Combining (11) and (16), the value of  $\beta$  in IGMP can be related to  $\alpha_1$  in ARP as

$$\beta = -\ln \frac{\alpha_1}{\Delta t}. \quad (17)$$

In this study, we call this process "combined IGMP-ARP" model. The main advantage of the model is that the inaccurate ACF does not have a detrimental effect on the estimated

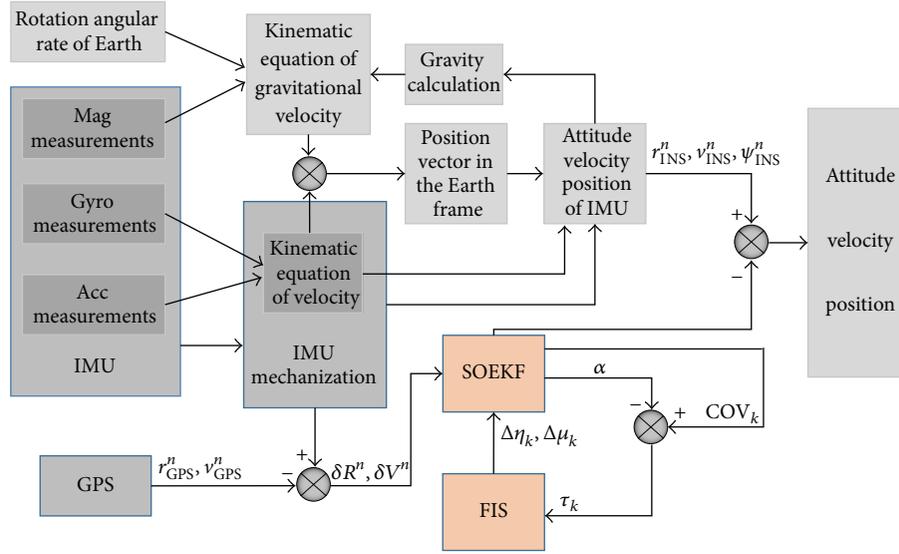


FIGURE 3: Proposed hybrid FIS-SOEKF model in navigation system.

correlation time. As it has been mentioned before, using the identification algorithms (i.e., YW, Burge's, and MCOV algorithms) to estimate the ARP's coefficients of the combined IGMP-ARP model is the goal of this section. Furthermore, in order to compare the combined IGMP-ARP model with classical IGMP model, first the IGMP parameters are experimentally estimated by ACF; then the parameters are obtained by ARP using different identification algorithms.

Furthermore, Wavelet Denoising Technique (WDT) can be combined with the ACF (as the experimental method to estimate the classical IGMP parameters) or with proposed combined IGMP-ARP (as an alternative method to estimate the parameters of IGMP) to investigate the stochastic error of inertial sensors. After applying the technique, the coefficients of IGMP obtained by ARP, or ACF, were determined from the enduring noises. These combinations are augmented in GPS/INS integration system to improve the navigational solution and their results are presented in Section 4.

**3.2. Proposed Hybrid FIS-SOEKF Model for GPS/INS Integration.** Two different approaches of adaptive Kalman filtering, namely, Innovation Adaptive Estimation (IAE) and Multiple Model Adaptive Estimation (MMAE), are considered by several research groups [32, 33]. These two approaches are based on Covariance Matching Estimation Technique (COMET). This study adopted the IAE concept in the fuzzy part of the proposed model.

The dynamic characteristics of vehicle motion in body frame and the navigation frame form the basis for the SOEKF process. The FIS can be exploited to increase the accuracy and the robustness of the SOEKF and to prevent its divergence in the tuning phase of SOEKF. Hence, FIS was used as a structure for identifying the dynamic variations and for implementing the real-time tuning of the nonlinear error model. It can provide a good estimation in maintaining the accuracy and

the tracking-capability of the system. Figure 3 depicts how the proposed hybrid FIS-SOEKF model performs as the ubiquitous navigation system.

Fuzzy Inference System (FIS) is a rule-based expert method that can mimic human thinking and understand linguistic concepts, rather than the typical logic systems [34]. The advantage of the FIS is realized when the algorithm of the estimation states becomes unstable because of the high complexity of the system. FIS are also used for the knowledge induction process, because they can serve as estimators for general purpose [35].

FIS architecture performs three types of operations: fuzzification, fuzzy inference, and defuzzification. Fuzzification converts the crisp input values into fuzzy values; fuzzy inference maps the given inputs into an output, and defuzzification converts the fuzzy operation into the new crisp values. The FIS can convert the inaccurate data to normalized fuzzy crisps, which are represented by the ranges of possible values, Membership Functions (MFs), and the confidence-rate of the inputs. In addition, the FIS are capable of choosing an optimal MF under certain specific criteria, applicable to a specific application. The deterministic output of FIS and its performance depend on the effective fuzzy rules, the considered defuzzification process, and the reliability of the MF values.

The proposed FIS model is based on the innovation process of the covariance matrix as the input of the FIS as well as the difference between the actual covariance matrix and the theoretical covariance matrix. Figure 4 shows the proposed FIS overview, which was used in this study. The theoretical covariance matrix, based on the innovation process, was computed partly in SOEKF by (4). The actual covariance matrix, according to [36], presented is proposed by

$$\alpha = \frac{1}{D} \sum_{i=k-D+1}^k v_i v_i^T, \quad (18)$$

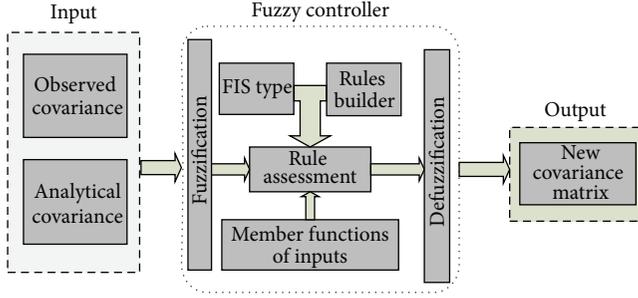


FIGURE 4: Proposed FIS part for hybrid FIS-SOEFK in navigation system.

where  $D$  is the window size (it is designated by moving window techniques and experimentally chosen as  $D = 20$ ) and  $v_i$  is  $[v_1 \ v_2 \ \dots \ v_m]^T$ . So, the difference is presented by

$$\begin{aligned} \tau = \text{COV}_k - \alpha &= \frac{1}{D} \\ &\cdot \sum_{i=k-D+1}^k \left\{ H_x(x_k^-, k) P_k^- H_x^T(x_k^-, k) \right. \\ &+ \frac{1}{2} \sum_{i,i'} e_i e_{i'}^T \text{tr} \left\{ H_{xx}^{(i)}(x_k^-, k) P_k^- H_{xx}^{(i')} x_k^- k P_k \right\} + \mu_k \\ &\left. - v_k v_k^T \right\}. \end{aligned} \quad (19)$$

In fact,  $\tau$  can display the Degree of Incompatibility (DOI) between the actual and the theoretical covariance matrices. When  $\tau$  is near zero, it means that these two values nearly match and the absolute value of the difference can be negligible. However, if  $\tau$  is smaller or greater than zero, it means that the theoretical value ( $\text{COV}_k$ ) is greater or smaller than the actual value ( $\alpha$ ). To correct this difference, the diagonal element of  $\mu_k$  in (4) should be adjusted relying on the DOI. The proposed rules of assessment, according to the difference between the actual and the theoretical covariance matrices, are described as three scenarios of FMs:

- (1) If  $\tau$  is greater than 0, then  $\mu_k$  will decrease due to  $\delta\mu_k$ .
- (2) If  $\tau$  is less than 0, then  $\mu_k$  will increase due to  $\delta\mu_k$ .
- (3) If  $\tau \cong 0$ , then  $\mu_k$  will remain unchanged.

Consider  $\delta\mu_k = \mu_k - \mu_{k-1}$ . The related FMs are shown in Figure 5(a), wherein “D,” “B,” and “I” denote “Decrease,” “Balance,” and “Increase,” respectively. In addition, (2) and (5) show that if  $\mu_k$  is perfectly observed, variation in  $\eta_k$  can make changes in  $\text{COV}_k$ , directly. So, with the observation of the incompatibility between the actual and the theoretical covariance matrices, augmentation, or diminishment becomes essential for  $\eta_k$ . The proposed rules of assessment for this purpose are described as two MFs of FIS:

- (1) If  $\tau_k - \tau_{k-1} = 0$ , then  $\eta_k$  will remain unchanged.
- (2) If  $\tau_k - \tau_{k-1} \neq 0$ , then  $\eta_k$  will change due to  $\delta\eta_k$ .

TABLE 1: Technical specification of the angular rate and the acceleration sensors in micro-iBB.

Accelerometer		Gyroscope	
Data rate	200 Hz	Data rate	200 Hz
Temperature range	-40~+85°C	Temperature range	-40~+85°C
Input range	±2 g	Input range	±250°/sec
Linear sensitivity	2 mg/LSB	Sensitivity	m°/sec/digit
Acceleration noise density	220 $\mu\text{g}/\sqrt{\text{Hz}}$	Rate noise density	0.03°/sec/ $\sqrt{\text{Hz}}$
		Nonlinearity	0.2%/sec

Consider  $\delta\eta_k = \eta_k - \eta_{k-1}$ . The related FMs are shown in Figure 5(b) wherein “LZ,” “Z,” and “GZ” denote “lower than zero,” “zero,” and “greater than zero,” respectively.

#### 4. Experimental Analysis, Results, and Discussion

The static raw data for analysis of the stochastic error was obtained from the wireless Micro-intelligent Black Box (Micro-iBB), which was designed and assembled by the VTADS team of the LASSENA Lab in the ETS. Micro-iBB consists of a triad of accelerometers, gyroscopes, magnetometers, and a temperature sensor. The serial numbers of the accelerometer, the gyroscope, and the demonstration board are LSM303DLHC, L3GD20, and STEVAL-MKI119V1, respectively. The selected aiding devices in Micro-iBB consist of a GNSS receiver (u-blox 7). Table 1 and Figure 6 present, respectively, the details of the Micro-iBB and its technical specification, which were used in this paper.

**4.1. Stochastic Error Modeling Technique.** This section analyzes the stochastic error to determine the effect of random errors on gyroscopes and accelerometers of the Micro-iBB. First, the ACFs were considered after recording the raw data. WDT was applied before data processing to weaken high frequency noises and remove preliminary biases in the sensors. By using WDT, uncorrelated and colored noises in the signal could be reduced. Figure 7 depicts the effect of different LODs in eliminating the system’s noise. Then, the related parameters of ACF for IGMP, namely,  $\sigma_{wk}^2$  and  $\beta$ , should be extracted. Figure 8(a) shows the ACF for accelerometers after using WDT with six LODs. This figure confirms the existence of residual noise along  $y$ -axis and  $z$ -axis of the accelerometers, even after applying the WDT. This noise is the uncorrelated term of the noise, which cannot be attenuated by the ACF in IGMP. The ACF of the  $x$ -axis of the accelerometer implies that the  $x$ -axis is more correlated than are the other axes.

Figure 8(b) shows the ACF of gyroscopes after using WDT with 6 LODs. This figure shows that although the  $y$ -axis and  $z$ -axis gyroscopes were influenced principally by HF and white noises, the  $x$ -axis of the gyroscope shows more correlated components than the other axes do. The comparison between Figures 1 and 8 clearly shows that

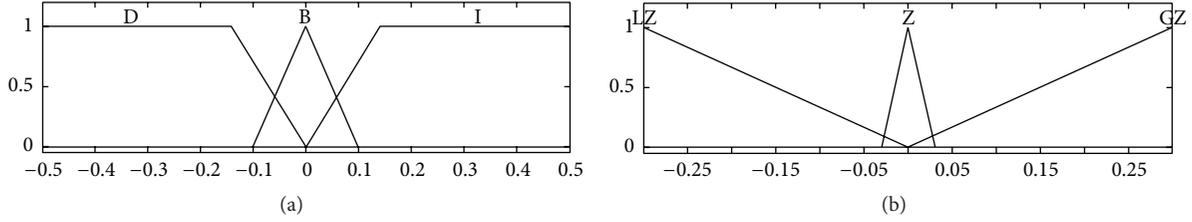


FIGURE 5: Membership Functions (MFs) for (a)  $\delta\mu_k$  and (b)  $\delta\eta_k$ .

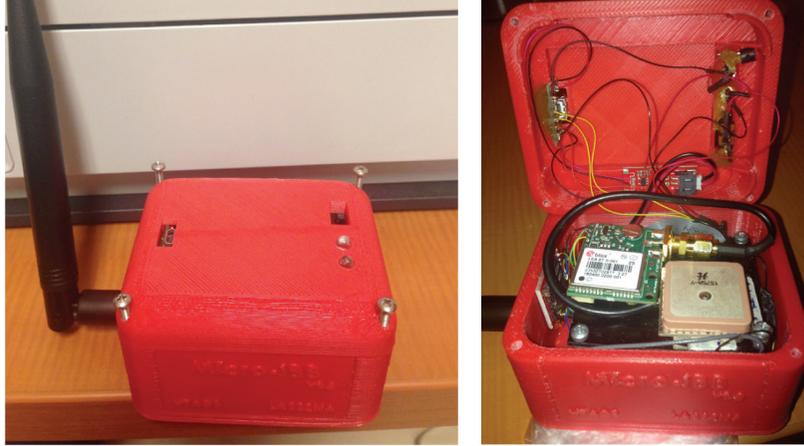


FIGURE 6: Wireless Micro-intelligent Black Box (Micro-iBB).

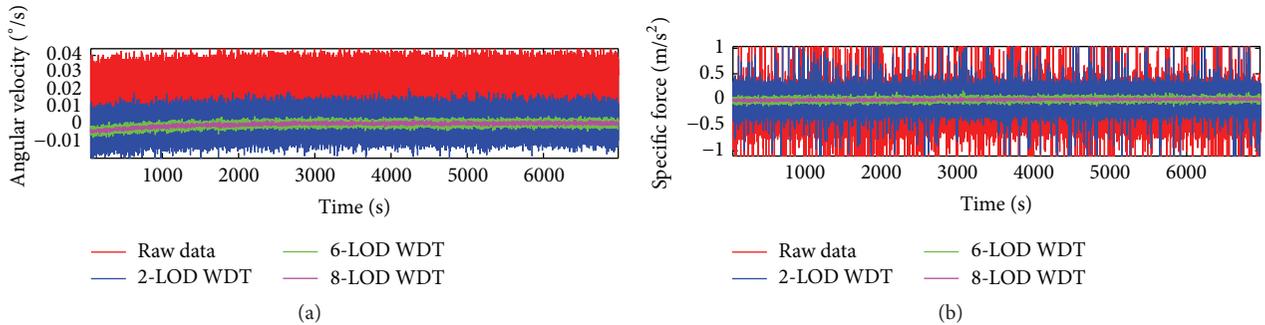


FIGURE 7: WDT with different LODs for (a)  $x$ -axis of accelerometers and (b)  $y$ -axis of gyroscopes, in Micro-iBB, in stationary situation.

the ACF cannot model the low-cost MEMS IMUs' errors with IGMP perfectly. Therefore, for this study, it was performed to model IGMP by utilizing ARP's parameters as an alternative solution to overcome this problem in the analysis of stochastic errors. The experimental results of ACF for all sensors are summarized in Table 2 under "ACF-based" column. Following this, MCOV, YW, and Burg's algorithms have also been applied to the denoised measurements (with different levels of decomposition) in order to estimate the parameters of combined IGMP-ARP. Results for these algorithms also are summarized in Table 2 under "YW-ARP-based," "MCOV-ARP-based," and "BURG's-ARP-based" columns for 6 levels of decomposition. These results present a considerable difference between the parameters obtained with each of the four identifications methods.

4.2. *GPS/INS Integration.* To evaluate the performance of the proposed models of GPS/INS integration, their results are compared with those of the conventional methods of GPS/INS integration. For demonstration, the loosely coupled GPS/INS integration was utilized in a three-dimensional navigation system. In the loosely coupled GPS/INS integration, position and velocity observed by GPS are the auxiliary measurements. So, in the SOEKF measurement model can be as follows:

$$\delta z = H\delta\bar{x} + \eta, \quad (20)$$

where  $H$  is  $\begin{bmatrix} H_R & 0_{3 \times 3} & 0_{3 \times 2} \\ 0_{3 \times 3} & H_R & 0_{3 \times 2} \end{bmatrix}$  and  $H_R = I_{3 \times 3}$ . Measurement noise vector  $\eta$  is  $[\eta_R \ \eta_V]^T$ , denoting the position and velocity noise. Vector  $\eta$  is defined by measurement noise model  $R$  in

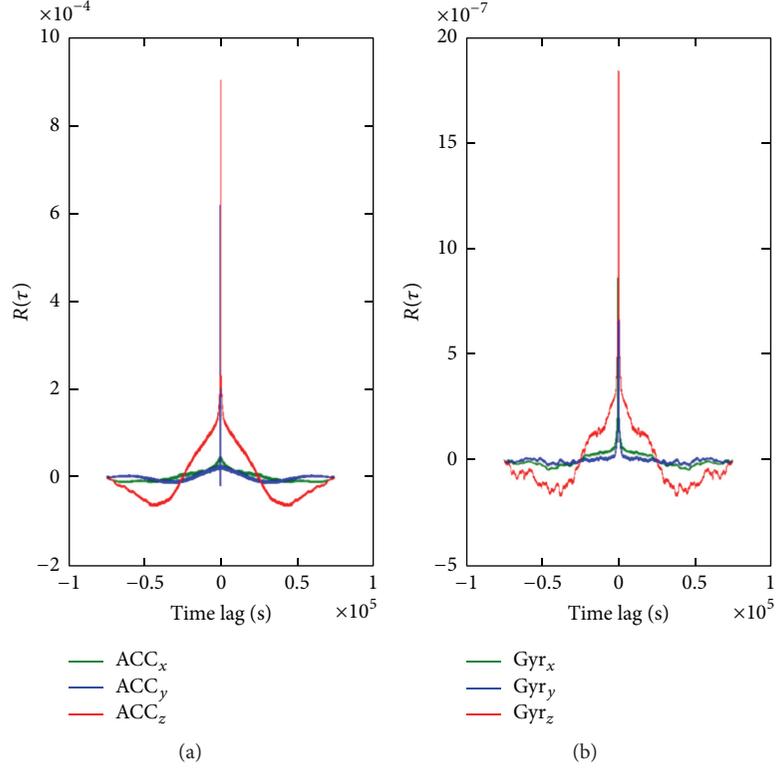


FIGURE 8: ACF of 1GMP for (a) accelerometers and (b) gyroscopes, in Micro-iBB after employing WDT with 6 LODs.

TABLE 2: The parameters needed to model 1GMP with different methods for ARP after 6 LODs of WTD.

Models	ACF-based		YW-ARP-based		MCOV-ARP-based		BURG's-ARP-based	
	$\sigma_{wk}^2$	$\beta$	$\sigma_{wk}^2$	$\beta$	$\sigma_{wk}^2$	$\beta$	$\sigma_{wk}^2$	$\beta$
Acc <sub>x</sub>	$3.2 \times 10^{-12}$	1/2600	$3.4 \times 10^{-10}$	1/380	$1.4 \times 10^{-11}$	1/740	$1.6 \times 10^{-11}$	1/1100
Acc <sub>y</sub>	$2.9 \times 10^{-11}$	1/2300	$3.8 \times 10^{-9}$	1/740	$1.8 \times 10^{-11}$	1/1100	$3.1 \times 10^{-11}$	1/1500
Acc <sub>z</sub>	$7.1 \times 10^{-12}$	1/1100	$5.4 \times 10^{-10}$	1/320	$4.6 \times 10^{-11}$	1/630	$5.1 \times 10^{-11}$	1/900
Gyr <sub>x</sub>	$2.5 \times 10^{-12}$	1/1700	$4.1 \times 10^{-10}$	1/60	$3.9 \times 10^{-11}$	1/520	$6.2 \times 10^{-11}$	1/1600
Gyr <sub>y</sub>	$1.8 \times 10^{-12}$	1/800	$6.7 \times 10^{-10}$	1/150	$2.5 \times 10^{-11}$	1/330	$4.1 \times 10^{-11}$	1/500
Gyr <sub>z</sub>	$3.7 \times 10^{-12}$	1/1500	$4.3 \times 10^{-10}$	1/90	$2.1 \times 10^{-11}$	1/570	$3.7 \times 10^{-11}$	1/1200

SOEKF, where  $R = \text{diag} [\delta_R^2 \delta_R^2 \delta_R^2 \delta_V^2 \delta_V^2 \delta_V^2]$  is modeled according to the GPS position and velocity uncertainty with zero mean and covariance matrix. The error states vector, relating to the navigation system of this study, is as follows:

$$\delta \bar{x} = [\delta \bar{R} \quad \delta \bar{V} \quad \delta \bar{\psi} \quad \delta \bar{b}_g \quad \delta \bar{b}_a]^T, \quad (21)$$

where  $\delta \bar{R}$  and  $\delta \bar{V}$  are, respectively,  $[\delta R_N \quad \delta R_E \quad \delta R_U]$  and  $[\delta V_N \quad \delta V_E \quad \delta V_U]$ , denoting position and velocity residual errors in East North Up (ENU) frame;  $\delta \bar{\psi}$  is  $[\delta \psi_N \quad \delta \psi_E \quad \delta \psi_U]$  and is the attitude error.  $\delta \bar{b}_g$  and  $\delta \bar{b}_a$  are, respectively,  $[\delta b_{g_x} \quad \delta b_{g_y} \quad \delta b_{g_z}]$  and  $[\delta b_{a_x} \quad \delta b_{a_y} \quad \delta b_{a_z}]$ , denoting the stochastic error states of accelerometers and gyroscopes along the three axes. In fact,  $\delta \bar{b}_g$  and  $\delta \bar{b}_a$  are included in the state vector of error propagation model to dynamically estimate the six stochastic error states. From the definition of 1GMP

in (10), the dynamic equations related to stochastic error states of accelerometers and gyroscopes can be shown by

$$\begin{bmatrix} \delta \dot{b}_g \\ \delta \dot{b}_a \end{bmatrix} = \begin{bmatrix} \beta_g & 0 \\ 0 & \beta_a \end{bmatrix} \begin{bmatrix} \delta \bar{b}_g \\ \delta \bar{b}_a \end{bmatrix} + \begin{bmatrix} w_g \\ w_a \end{bmatrix}. \quad (22)$$

Table 2 presents the 1GMP's parameters which are estimated by ACF and the combined 1GMP-ARP (using different algorithms) for 6 LODs of WTD. So, the  $\beta_g$ ,  $\beta_a$ ,  $w_g$ , and  $w_a$  in (21) can be initialized by the values in Table 2 for 6 LODs of WTD.

The proposed land-navigation solution was implemented with an experimental setup for a road test. The tests were performed by Micro-iBB, whose specifications are detailed under Section 4.1. The results were evaluated against a reference solution provided by Novatel SPAN technology. It consists of Novatel receiver and Honeywell tactical-grad IMU

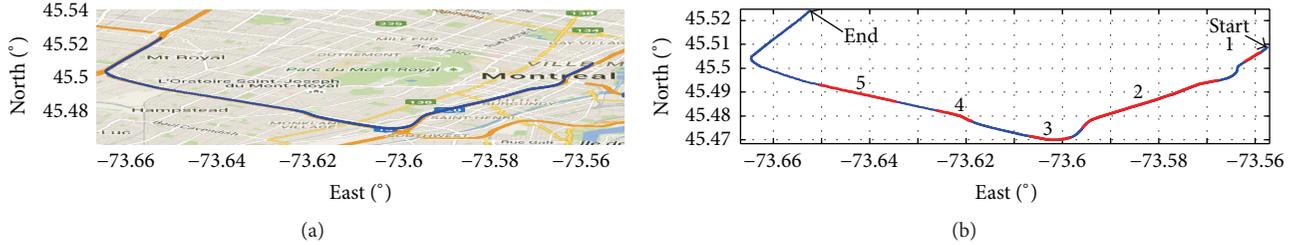


FIGURE 9: The freeway trajectory showing the five outages: (a) on google map and (b) on Matlab.

to validate the proposed method and to assess the overall performance during the GPS outages.

The raw data of the route-test was gathered from different sensors of Micro-iBB. The selected trajectories satisfy the overall quality and reliability requirements of the proposed models of this study in real environments of various conditions. The first trajectory was figured out in an urban freeway which allowed the authors to drive at high speed. The second trajectory was in downtown area that contains numerous prominent skyscrapers. The test along this trajectory was performed at slow speed with frequent stops because of high traffic and crowded road intersections.

The goal is to investigate the efficiency of the proposed hybrid FIS-SOEKF using different LODs of WDT and using various algorithms to estimate ARP's coefficients for IGMP. The WDTs considered for this purpose are 2 LODs, 6 LODs, and 8 LODs and the algorithms are MCOV, YW, and Burg's algorithm. The proposed combinations are compared with traditional SOEKF and two hybrid FIS-SOEKFs, which used ACF to estimate the IGMP coefficient with different LODs of WDT. All the hybrid FIS-SOEKF solutions followed the idea of updating stochastic error states to the gyroscope and accelerometers measurements. RMSEs were estimated in all navigational solutions with Novatel SPAN as the reference solution.

**4.2.1. First Scenario: Freeway.** The first route-test trajectory selected for this study starts from freeway #5 near Viger East Avenue (45.509023, -73.557511) and ends at freeway #71 near Lebeau Avenue (45.524299, -73.652472) in Montreal, QC, Canada, by boulevard Ville-Marie/highway #720 West and highway #15 North (see Figure 9). This trajectory was done for nearly 11 min of uninterrupted car navigation over a distance of 14.4 km. The freeway trajectory was chosen with five natural GPS outages to examine the system's performance during short and long outages that exist in urban canyons. Figure 9(a) depicts the route-test with the reference solution on google map. Figure 9(b) presents the position of the GPS outages on Matlab with red lines, along the selected trajectory.

Figure 10 presents the number of available satellites and the time-duration of each GPS outage. The first GPS outage in tunnel area occurred for 30 sec with less than number of 7 available satellites. There was no satellite for about two-thirds of the duration of this outage (see Figure 10(a)). The second and third outages occurred along the two skyways for 180 sec and 60 sec durations (see Figures 10(b)-10(c)). Figures

10(d)-10(e) present the last two outages in urban canyon area for durations of 50 and 30 sec. Generally, higher moving speed increases horizontal positioning errors. Therefore, this scenario of high speed driving with several outages can test the validity and robustness of the proposed models.

Figure 11 presents the root mean square and the maximum errors in horizontal positioning during the five GPS outages in the first scenario for the twelve compared solutions. The advantage of using the proposed method with 6 LODs for updating stochastic error states to gyroscope and accelerometer measurements can be seen by comparing it with the traditional SOEKF and hybrid FIS-SOEKFs, which are used in ARP or ACF to estimate the IGMP coefficient, utilizing different LODs of WDT. Figure 11 illustrates the results obtained by (i) the traditional SOEKF without updating stochastic error states, shown as SOEKF, and (ii) those obtained by the proposed hybrid FIS-SOEKF, using ACF-based IGMP, after 2-LOD denoising, shown as "FS(ACF2)," and after 6-LOD denoising, shown as "FS(ACF6)." The rest of the solutions used the proposed hybrid FIS-SOEKF and are shown as "FS" in this figure. (iii) "MCOV-ARP," "YW-ARP," and "BURG's-ARP" represent the solutions that modeled IGMP by using ARP's coefficients under "modified-covariance," "Yule-Walker," and "Burg's" algorithms, respectively. The values, which follow each solution, denote the specific LOD of WDT that was used in that solution.

From the foregoing data, it can be seen that the proposed hybrid FIS-SOEKF, which used ACF-based IGMP after 6-LOD denoising, performed significantly better than did the traditional SOEKF without updating stochastic error states. This is because the proposed hybrid FIS-SOEKF can deal with the nonlinearity of systems and models; however, SOEKF exploits second-order linearized models to estimate the state of systems. The results also show that "BURG's-ARP" outperformed "SOEKF" and also the proposed hybrid FIS-SOEKF that used ACF-based IGMP after denoising.

Furthermore, in terms of the ARP-based IGMP after 6-LOD denoising, "BURG's-ARP6" shows a major improvement in positioning accuracy as compared with the performance of "YW-ARP6" and "MCOV-ARP6." This is because Burg's algorithm provides the highest accuracy when applied to low-cost MEMS-based INS as compared to the other algorithms that determine the ARP's coefficients ideally. During the second outage of 180 sec duration, it can be seen that the performance of all solutions degraded considerably, because at least four satellites are needed in loosely coupled

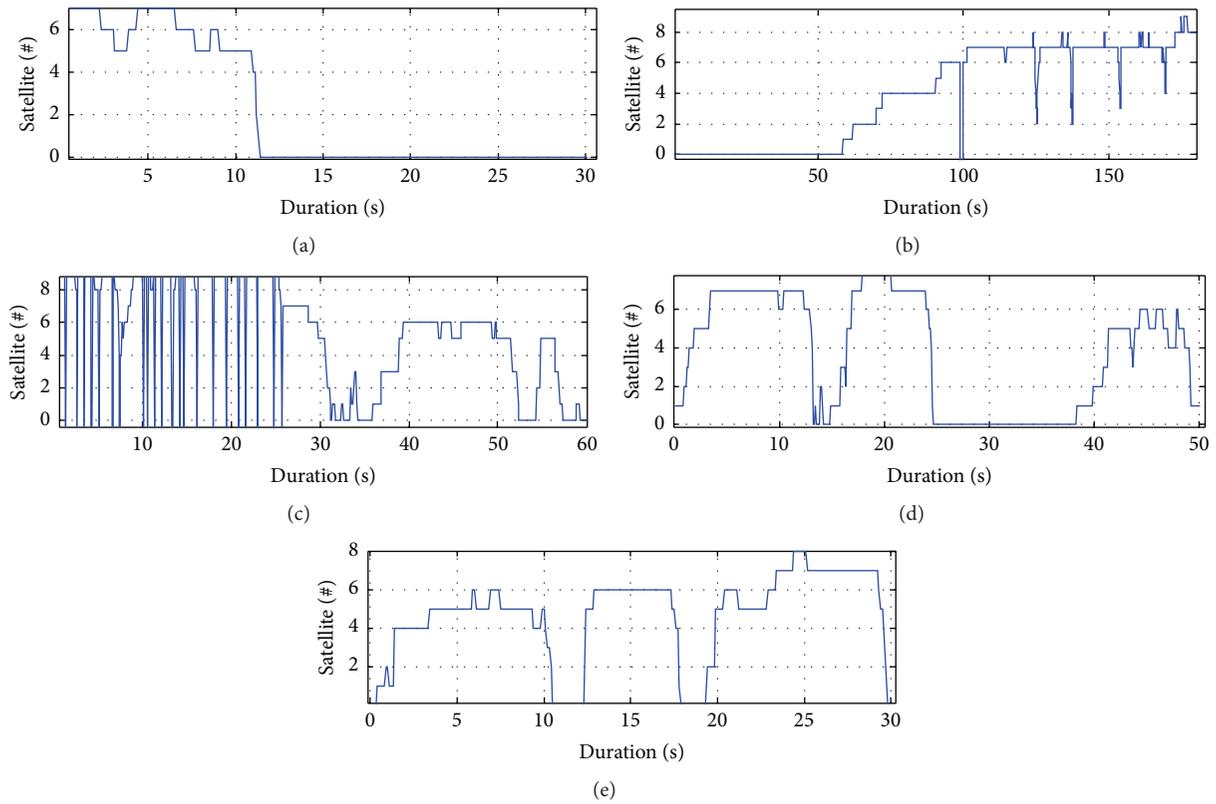


FIGURE 10: Available satellites in each outage for freeway trajectory: (a) outage #1, (b) outage #2, (c) outage #3, (d) outage #4, and (e) outage #5.

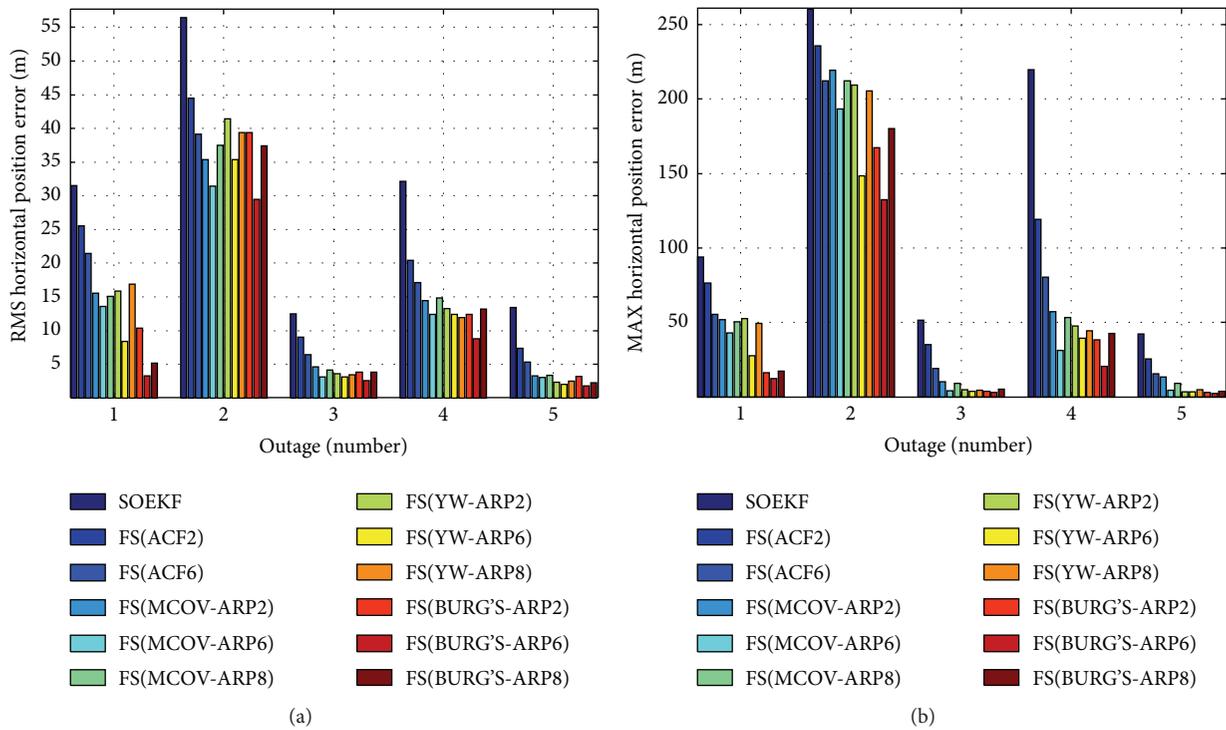


FIGURE 11: (a) Root mean square error and (b) maximum errors, in horizontal positioning during five GPS outages in the first scenario (freeway).

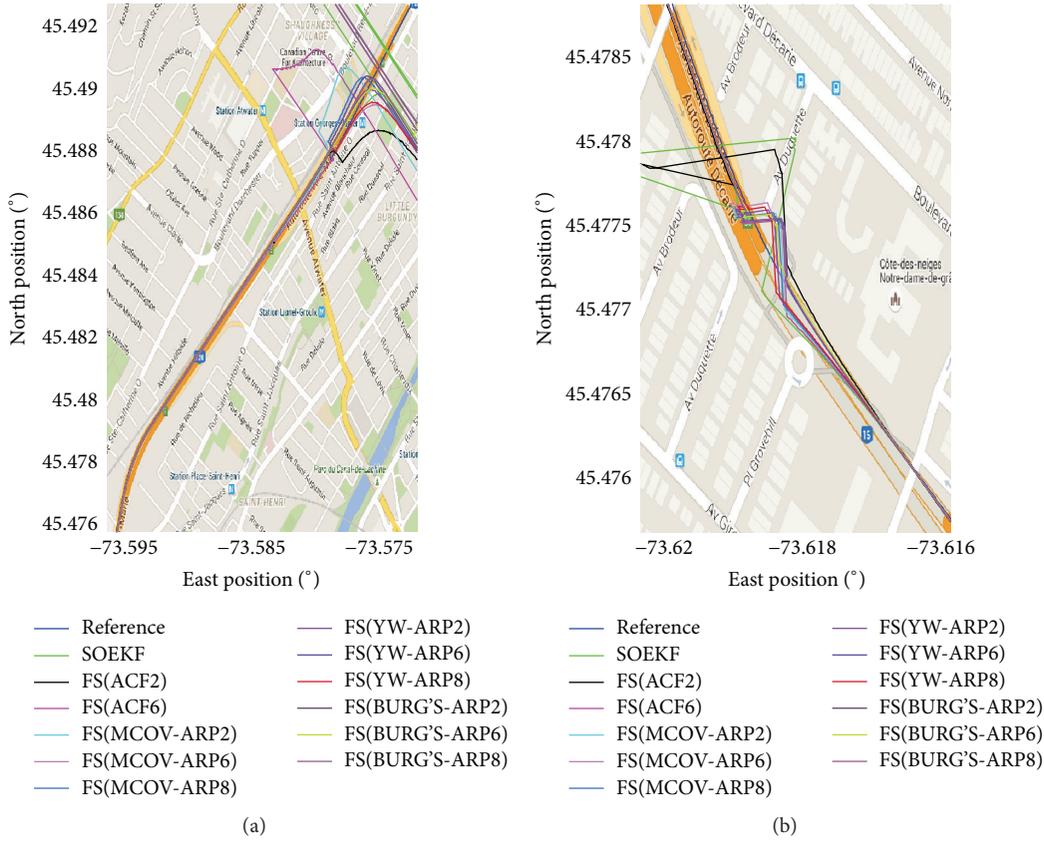


FIGURE 12: Different navigation solutions for the freeway trajectory: (a) outage #2 and (b) outage #4.

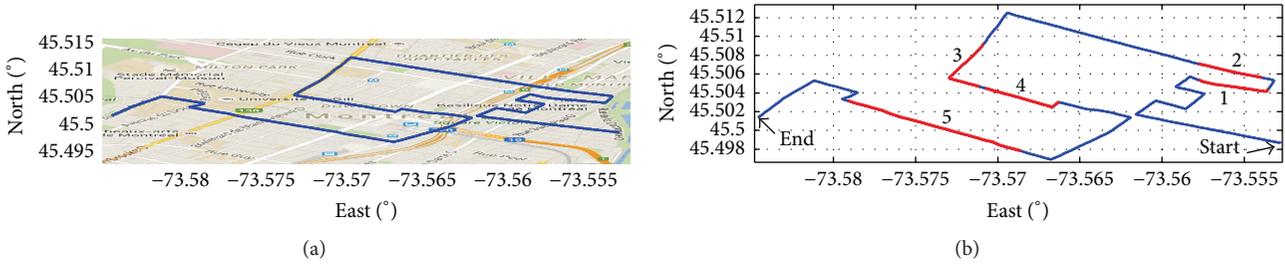


FIGURE 13: Second (downtown) trajectory showing the five outages: (a) on google map and (b) on Matlab.

GPS/INS integration. Against this requirement, the number of satellites available was less than four for more than half of the outage duration, besides the duration itself being quite long. In addition, using the low-cost MEMS-based INS worsened this situation. So, the second outage gave the worst result, with the maximum horizontal error being 630 m and mean horizontal error 187 m; these cannot be shown in the figure as the values are high.

Figure 12 presents two sections of the freeway trajectory during GPS outages #2 and #4. This figure shows the reference and all the navigation solutions considered in this paper. These results confirm that the performance of the proposed hybrid FIS-SOEKF, used with ARP-based IGMP after 6-LOD denoising, is acceptable as a new navigational solution

in freeways where the duration of GPS outages can be a maximum of 60 sec.

4.2.2. *Second Scenario: Downtown.* The second route-test trajectory selected starts from McGill Street near La Commune Ouest Avenue (45.498718, -73.552831) and ends up at Pins Ouest Avenue (45.501466, -73.584548) in Montreal, QC, Canada, by Boulevard St-Laurent/Shebrooke Ouest Street and Peel Street (see Figure 13). Testing along this trajectory was carried out through medium and high traffic volumes for nearly 50 min of car navigation covering a distance of 10.5 km. This downtown trajectory, which runs through skyscrapers, was chosen because of its five natural GPS outages to examine the system's performance during short and long outages.

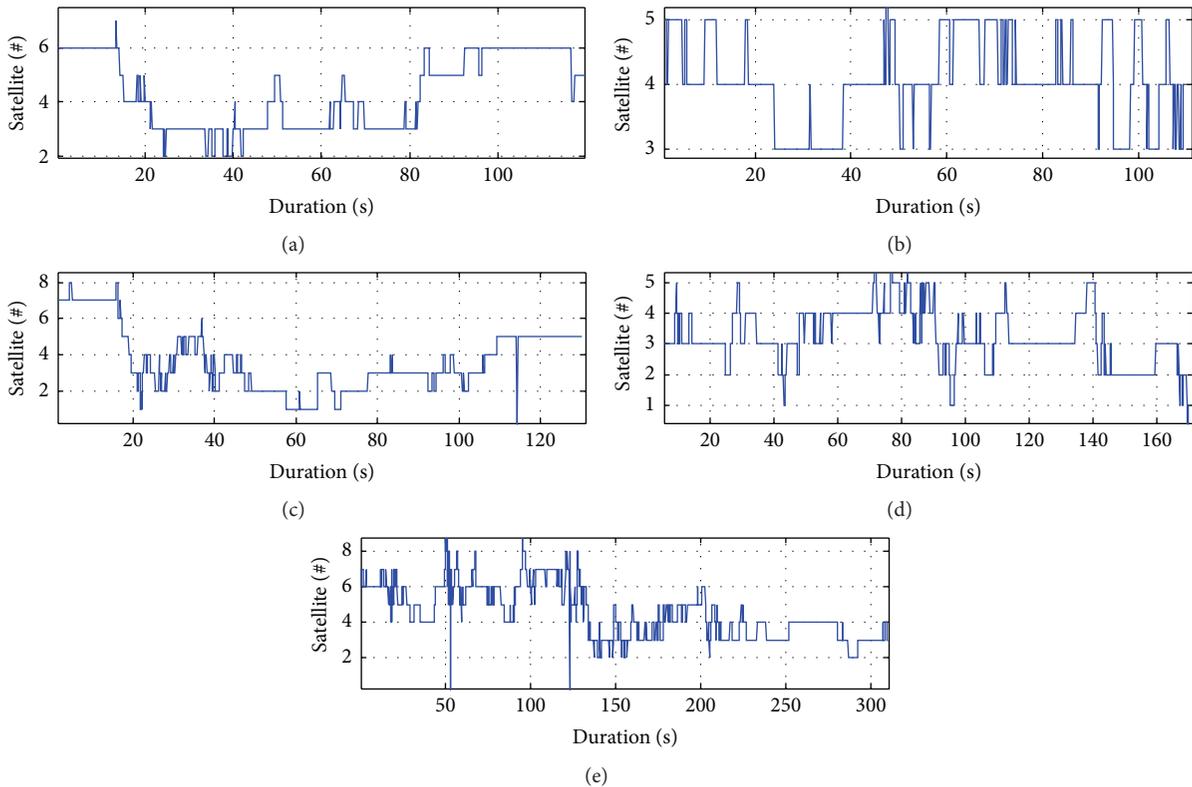


FIGURE 14: Available number of satellites during each outage for downtown trajectory: (a) outage #1, (b) outage #2, (c) outage #3, (d) outage #4, and (e) outage #5.

Figure 13(a) depicts route-test with the reference solution on google map. Figure 13(b) presents the details of the position of the GPS outages on Matlab with red lines, along the selected trajectory.

Figure 14 shows the number of available satellites and the time-duration of each GPS outage for downtown trajectory of Montreal. To examine the proposed navigation solution during longer GPS outages, the near outages were clubbed into bigger outages. The number of the bigger outages in this scenario thus became 5. Figure 15 shows the RMSE and the maximum error in horizontal positioning during the five GPS outages for the twelve compared solutions. The results of this scenario confirm the accuracy obtained in the results of the previous scenario.

Figure 16 illustrates the results obtained for outages #2 and #4, by (i) the traditional SOEKF without updating stochastic error states, shown as “SOEKF”, and (ii) those obtained by the proposed hybrid FIS-SOEKF using ACF-based 1GMP after 2-LOD denoising, shown as “FS(ACF2),” and after 6-LOD denoising, shown as “FS(ACF6).” (iii) The rest of the solutions used the proposed hybrid FIS-SOEKF which are shown as “FS.” “MCOV-ARP,” “YW-ARP,” and “BURG’s-ARP” represent the solutions that modeled 1GMP by using ARP’s coefficients under “modified-covariance,” “Yule-Walker,” and “Burg’s” algorithms, respectively. The values that follow each solution denote the specific LOD of WTD that was used in that solution.

From the data presented, it can be seen that the proposed hybrid FIS-SOEKF, which used ACF-based 1GMP after 2-LOD and 6-LOD denoising, “FS(ACF2)” and “FS(ACF6),” performed significantly better than did the traditional SOEKF without updating stochastic error states. The results also show that “FS(BURG’s-ARP),” among all the “SF (ARP)” navigation solutions, outperformed “SOEKF,” “FS(ACF2),” and “FS(ACF6).” Furthermore, in terms of the ARP-based 1GMP after 6-LOD denoising, “FS(BURG’s-ARP6)” shows a major improvement in positioning accuracy in the downtown scenario, as compared with the accuracy obtained by “FS(YW-ARP6)” and “FS(MCOV-ARP6).” This is because Burg’s algorithm provides the highest accuracy when applied to low-cost MEMS-based INS, as compared to the other algorithms that determine the ARP’s coefficients perfectly.

As regards the number of satellites and the duration of each GPS outage, it can be seen that there are less than 4 satellites for two-thirds of the duration and less than 5 satellites during the remaining part of the duration for outages #2 and #4. However, RMSE in outage #4 is much more than that in outage #2, because the duration of outage #4 (180 sec) is longer than that of outage #2 (110 sec). The results also show that outage #5 presents, in spite of its longest duration (310 sec), lesser RMSE than that of outages #3 and #4. This is because 6–8 satellites existed for less than half of the duration of outage #4 and less than 4 satellites for three-quarters of the duration. Outage #2 shows the lowest error (2 m) because of its short outage duration.

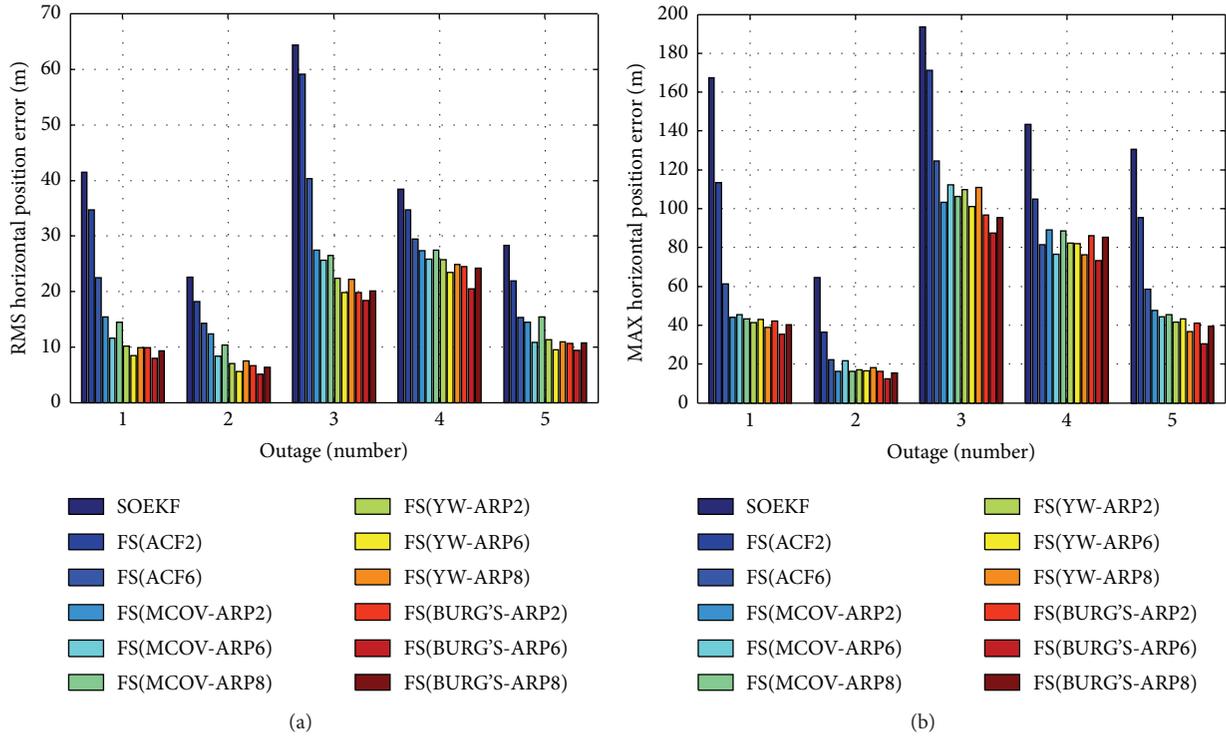


FIGURE 15: Horizontal positioning errors during the five GPS outages in the downtown scenario: (a) root mean square error and (b) maximum errors.

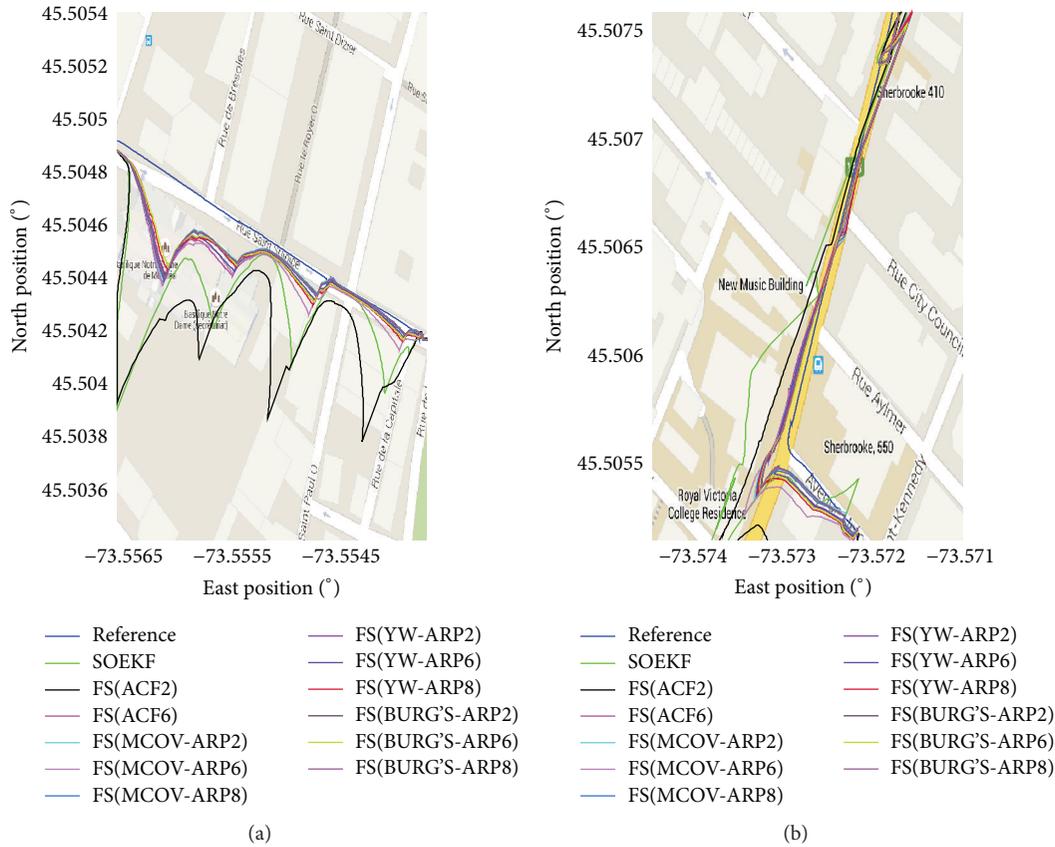


FIGURE 16: Different navigation solutions for downtown trajectory: (a) outage #2 and (b) outage #4.

## 5. Conclusions

In this paper, two novel auxiliary methods are proposed to improve the performance of ultralow-cost MEMS-based IMU in a vehicular navigation system. The proposed methods can cover two areas: modeling stochastic errors and performing GPS/INS integration. The proposed IGMP-ARP consists of a complete stochastic error modeling of the noise components in MEMS low-cost inertial sensors. This paper shows a big cooperation between IGMP, ARP, and WDT to characterize the noise components. Different levels of decomposition in denosing technique as well as various methods to identify the ARP parameters were considered to clarify the best one for stochastic error modeling in MEMS low-cost inertial sensors. Result presents that Burg's method, after applying the six levels of decomposition in denosing technique, performs much better than modified-covariance and Yule-Walker methods with different levels of decomposition. The second proposed method consists of an integration of SOEKFFIS to enhance the performance of low-cost GPS/INS integration for navigation data fusion. The FIS part is exploited for dynamic adjustment of the process noise covariance, by observing the innovation process, which is utilized in SOEKF-part to maintain further enhancement in the estimation of the accuracy. The results confirm that the proposed hybrid FIS-SOEKF can provide further improvement in the overall performance compared to SOEKF in harsh environment.

Future research work related to this study will focus on more complicated GPS/INS integration structures such as tightly and ultratightly coupled and noise modeling with higher order stochastic error to reduce the errors in the positing of the navigation system. In addition, GPS/INS integration will expand to GNSS/BDS/low-cost-INS integration to utilize the advantages of the multi-GNSS environment.

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

This research is part of the project entitled VTADS: Vehicle Tracking and Accident Diagnostic System. It is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and École de Technologie Supérieure (LASSENA Lab), in collaboration with two industrial partners, namely, iMetrik Global Inc. and Future Electronics.

## References

- [1] J. W. Zellner, R. M. van Auken, D. P. Chiang, P. C. Broen, J. Kelly, and Y. Sugimoto, "Extension of the Honda-DRI 'Safety Impact Methodology' (SIM) for the NHTSA Advanced Crash Avoidance Technology (ACAT) program and application to a prototype advanced collision mitigation braking system," *SAE International Journal of Passenger Cars—Mechanical Systems*, vol. 2, no. 1, pp. 875–894, 2009.
- [2] T. Hai Ta, D. Minh Truong, T. Thanh Thi, H. Nguyen, T. Dinh Nguyen, and G. Belforte, "Multi-GNSS positioning campaign in South-East Asia," *Coordinates*, vol. 9, pp. 11–20, 2013.
- [3] D. M. Truong and T. H. Ta, "Development of real multi-GNSS positioning solutions and performance analyses," in *Proceedings of the International Conference on Advanced Technologies for Communications (ATC '13)*, pp. 158–163, Ho Chi Minh, Vietnam, October 2013.
- [4] K.-P. Schwarz and N. El-Sheimy, "Future positioning and navigation technologies," in *Study Performed under the Scientific Services Agreement with Batelle*, Columbus Division and Topographic Engineering Center, Fort Belvoir, VA, USA, 1999.
- [5] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*, John Wiley & Sons, New York, NY, USA, 2007.
- [6] J. Georgy, U. Iqbal, and A. Noureldin, "Quantitative comparison between kalman filter and particle filter for low cost INS/GPS integration," in *Proceedings of the 6th International Symposium on Mechatronics and its Applications (ISMA '09)*, pp. 1–7, Sharjah, United Arab Emirates, March 2009.
- [7] A. Oppenheim and R. Schaffer, *Discrete-Time Signal Processing: Pearson New International Edition*, Pearson Education, Upper Saddle River, NJ, USA, 2013.
- [8] A. Noureldin, T. B. Karamat, M. D. Eberts, and A. El-Shafie, "Performance enhancement of MEMS-based INS/GPS integration for low-cost navigation applications," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1077–1096, 2009.
- [9] T. B. Karamat, *Implementation of tightly coupled INS/GPS integration for land vehicle navigation [M.S. thesis]*, Electrical engineering, Royal Military College of Canada, 2009.
- [10] J. Pusa, *Strapdown inertial navigation system aiding with non-holonomic constraints using indirect Kalman filtering [M.S. thesis]*, Department of Mathematics, Tampere University of Technology, 2009.
- [11] Z. Shen, J. Georgy, M. J. Korenberg, and A. Noureldin, "Low cost two dimension navigation using an augmented Kalman filter/Fast Orthogonal Search module for the integration of reduced inertial sensor system and Global Positioning System," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1111–1132, 2011.
- [12] L. Zhao, H. Qiu, and Y. Feng, "Study of robust  $H_\infty$  filtering application in loosely coupled INS/GPS system," *Mathematical Problems in Engineering*, vol. 2014, Article ID 904062, 10 pages, 2014.
- [13] P. Aggarwal, Z. Syed, and N. El-Sheimy, "Hybrid Extended Particle Filter (HEPF) for integrated civilian navigation system," in *Proceedings of the IEEE/ION Position, Location and Navigation Symposium (PLANS '08)*, pp. 984–992, Monterey, Calif, USA, May 2008.
- [14] J. Cheng, D. Chen, R. Landry Jr., L. Zhao, and D. Guan, "An adaptive unscented kalman filtering algorithm for MEMS/GPS integrated navigation systems," *Journal of Applied Mathematics*, vol. 2014, Article ID 451939, 8 pages, 2014.
- [15] K.-W. Chiang and Y.-W. Huang, "An intelligent navigator for seamless INS/GPS integrated land vehicle navigation applications," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 722–733, 2008.
- [16] A. Noureldin, A. El-Shafie, and M. R. Taha, "Optimizing neuro-fuzzy modules for data fusion of vehicular navigation systems using temporal cross-validation," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 1, pp. 49–61, 2007.

- [17] L. Semeniuk and A. Noureldin, "Modeling of INS position and velocity errors using radial basis function neural networks for INS/GPS integration," in *Proceedings of the ION National Technical Meeting*, pp. 18–20, Monterey, Calif, USA, January 2006.
- [18] A. Noureldin, A. El-Shafie, and M. Bayoumi, "GPS/INS integration utilizing dynamic neural networks for vehicular navigation," *Information Fusion*, vol. 12, no. 1, pp. 48–57, 2011.
- [19] J. Škaloud, *Optimizing georeferencing of airborne survey systems by INS/DGPS [Ph.D. thesis]*, Department of Geomatics Engineering, Alberta, Canada, 1999.
- [20] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*, John Wiley & Sons, New York, NY, USA, 2004.
- [21] B. Sadeghi and B. Moshiri, "Second-order EKF and unscented Kalman filter fusion for tracking maneuvering targets," in *Proceedings of the IEEE International Conference on Information Reuse and Integration (IEEE IRI '07)*, pp. 514–519, IEEE, Las Vegas, Nev, USA, August 2007.
- [22] R. Ardito, C. Comi, A. Corigliano, and A. Frangi, "Solid damping in micro electro mechanical systems," *Meccanica*, vol. 43, no. 4, pp. 419–428, 2008.
- [23] A. Kumar and D. Chakraborty, "Effective properties of thermo-electro-mechanically coupled piezoelectric fiber reinforced composites," *Materials & Design*, vol. 30, no. 4, pp. 1216–1222, 2009.
- [24] M. Kleiber and T. D. Hien, *The Stochastic Finite Element Method: Basic Perturbation Technique and Computer Implementation*, Wiley-Interscience, 1992.
- [25] M. Kamiński and A. Corigliano, "Sensitivity, probabilistic and stochastic analysis of the thermo-piezoelectric phenomena in solids by the stochastic perturbation technique," *Meccanica*, vol. 47, no. 4, pp. 877–891, 2012.
- [26] M. El-Diasty and S. Pagiatakis, "A rigorous temperature-dependent stochastic modelling and testing for MEMS-based inertial sensor errors," *Sensors*, vol. 9, no. 11, pp. 8473–8489, 2009.
- [27] P. Lavoie, *Système de Navigation Hybride GPS/INS à Faible Coût pour la Navigation Robuste en Environnement Urbain*, M. Ing., Génie Électrique, École de Technologie Supérieure, Montreal, Canada, 2012.
- [28] S. Nassar, K.-P. Schwarz, N. El-Sheimy, and A. Noureldin, "Modeling inertial sensor errors using autoregressive (AR) models," *Navigation*, vol. 51, no. 4, pp. 259–268, 2004.
- [29] S. Nassar, K.-P. Schwarz, N. El-Sheimy, and A. Noureldin, "Modeling inertial sensor errors using autoregressive (AR) models," *Navigation, Journal of the Institute of Navigation*, vol. 51, no. 4, pp. 259–268, 2004.
- [30] C. H. Kang, S. Y. Kim, and C. G. Park, "Improvement of a low cost MEMS inertial-GPS integrated system using wavelet denoising techniques," *International Journal of Aeronautical and Space Sciences*, vol. 12, no. 4, pp. 371–378, 2011.
- [31] N. El-Sheimy, S. Nassar, and A. Noureldin, "Wavelet de-noising for IMU alignment," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 10, pp. 32–39, 2004.
- [32] F. Jiancheng and Y. Sheng, "Study on innovation adaptive EKF for in-flight alignment of airborne POS," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 4, pp. 1378–1388, 2011.
- [33] J. Ma, S. Ni, W. Xie, and W. Dong, "An improved strong tracking multiple-model adaptive estimation: a fast diagnosis algorithm for aircraft actuator fault," *Transactions of the Institute of Measurement and Control*, 2015.
- [34] D. Bhatt, P. Aggarwal, V. Devabhaktuni, and P. Bhattacharya, "A new source difference artificial neural network for enhanced positioning accuracy," *Measurement Science and Technology*, vol. 23, no. 10, Article ID 105101, 2012.
- [35] L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, and M. E. Lopez, "Real-time system for monitoring driver vigilance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 63–77, 2006.
- [36] R. Vershynin, "How close is the sample covariance matrix to the actual covariance matrix?" *Journal of Theoretical Probability*, vol. 25, no. 3, pp. 655–686, 2012.

## Research Article

# Bayesian Train Localization with Particle Filter, Loosely Coupled GNSS, IMU, and a Track Map

**Oliver Heirich**

*DLR (German Aerospace Center), Institute of Communications and Navigation, 82234 Oberpfaffenhofen, Germany*

Correspondence should be addressed to Oliver Heirich; [oliver.heirich@dlr.de](mailto:oliver.heirich@dlr.de)

Received 23 October 2015; Revised 5 February 2016; Accepted 24 March 2016

Academic Editor: Yassine Ruichek

Copyright © 2016 Oliver Heirich. 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.

Train localization is safety-critical and therefore the approach requires a continuous availability and a track-selective accuracy. A probabilistic approach is followed up in order to cope with multiple sensors, measurement errors, imprecise information, and hidden variables as the topological position within the track network. The nonlinear estimation of the train localization posterior is addressed with a novel Rao-Blackwellized particle filter (RBPF) approach. There, embedded Kalman filters estimate certain linear state variables while the particle distribution can cope with the nonlinear cases of parallel tracks and switch scenarios. The train localization algorithm is further based on a track map and measurements from a Global Navigation Satellite System (GNSS) receiver and an inertial measurement unit (IMU). The GNSS integration is loosely coupled and the IMU integration is achieved without the common strapdown approach and suitable for low-cost IMUs. The implementation is evaluated with real measurements from a regional train at regular passenger service over 230 km of tracks with 107 split switches and parallel track scenarios of 58.5 km. The approach is analyzed with labeled data by means of ground truth of the traveled switch way. Track selectivity results reach 99.3% over parallel track scenarios and 97.2% of correctly resolved switch ways.

## 1. Introduction

Train localization inside a railway network is necessary for a collision-free operation and mainly addressed by centralized traffic control, signaling, and sensors in the railway infrastructure. Onboard train localization in combination with communications enables distributed and train centric assistant systems such as collision avoidance, coupling, and autonomous train operation. This localization system concept focuses on exclusive onboard computation and sensors without any additional railway infrastructure.

Future railway systems such as a train centric collision avoidance system [1, 2] require a localization system with continuous availability and a track-selective accuracy. Track selectivity is the ability to identify the correct track, especially in the critical, parallel track scenario after a ride on a divisive switch way. The track selectivity is the technical challenge and also the major requirement of train localization.

The goal of train localization is to determine the position of the train in the track network by topological coordinates, which are hidden variables and cannot be measured directly.

Single sensor systems, such as global navigation satellite systems (GNSS), are very beneficial for localization, but the measurement accuracy and lack of availability in parts of the railway environment do not fulfill the safety-critical requirements.

Research on train localization with onboard sensors focuses on the following question: how is a train-borne, safety-critical, and onboard localization system designed and analyzed in terms of data processing with continuous availability and a track-selective accuracy? The approach should cope with hidden variables as the topological position, imprecise information of measurements from multiple sensor sources, outages, statistical noise, and systematic measurement errors.

This paper presents a train localization approach by a Rao-Blackwellized particle filter (RBPF). Figure 1 shows the setup with onboard sensor data of an inertial measurement unit (IMU) and a GNSS receiver. The RBPF estimates the linear state variables of the one-dimensional train transition with Kalman filters within each particle of a particle filter. Furthermore, a novel empirical evaluation methodology

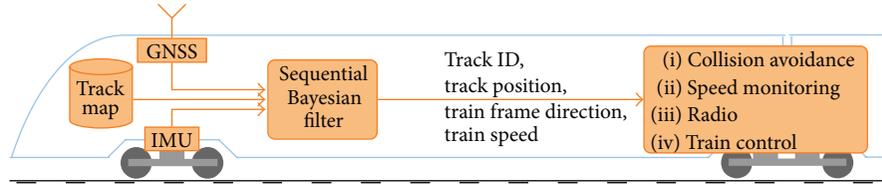


FIGURE 1: Bayesian train localization setup with GNSS, IMU, and a track map.

is defined for track selectivity which is not specific to a certain train localization approach. This paper is considered as a follow-up of the theoretic probabilistic approach with a particle filter [3–5] using satellite range measurements. The novel parts are the extension of the particle filter for train localization with a Rao-Blackwellization as well as an evaluation framework for track selectivity. The RBPF and a reference map-match approach are evaluated in terms of track-selective accuracy with real train data of a regional train.

The outline of this paper starts with a related work review, a general information for a map-based train localization in Section 3, and the description of the used sensors in Section 4. Section 5 contains the derivation of the Rao-Blackwellization and the RBPF implementation is given in Section 6. The track-selective evaluation (Section 7) evaluates the RBPF approach with data from train runs (Section 8). Sections 9 and 10 show the results and the discussions on results.

## 2. Related Work

There are multiple approaches of train localization in the literature with onboard sensors and a map. A selection of studies are chosen, which focus on track selectivity or specific sensor studies, which can identify the switch way. The different approaches vary in sensor types or combinations, processing methods, and evaluation scope and will be presented in these categories.

Inertial sensors are often used in combination with an integrated navigation system for GNSS position aiding, for example, in [6, 7]. The yaw turn rate can also be used for the switch way identification that has been used in [8, 9] and analyzed in [10, 11]. Approaches with GNSS and IMU are found in [7–9] and extensions with eddy current sensor in [6, 12, 13]. The eddy current sensor, in principle a metal detector for characteristic railway features, can be used for a switch way detection, as speed or displacement sensor. Sensors such as cameras [14, 15] or LIDAR [16, 17] can directly identify the different switch ways and contribute to the track-selective result. A study of a tightly coupled localization with raw GNSS data and a track map was shown in [5, 18]. A tightly coupled approach considers pseudorange, Doppler, and phase measurements and has the advantage to process location information even with less than four satellites in view. However, a tightly coupled approach is typically more complex to implement than a loosely coupled approach and a user clock offset needs to be estimated additionally. The processing methods of each study are different and dependent

on the used sensors, filtering method, and algorithmic integration of the map. Saab proposed a train localization using a map-matching technique with a correlation of the curvature signature [19]. For the track selectivity, there are two classes with map integration and estimation of importance: a multiple hypothesis filter handles and maintains multiple estimates on several tracks in the vicinity and is commonly used [6, 9, 13, 17]. The particle filter uses usually a large set of particles, which are location hypotheses on a map, and handles the different track hypotheses by particles. The particle filter with onboard sensors and a map was proposed in [20]. Fouque and Bonnifait [18] defined a marginalized particle filter with raw GNSS data for the identification of the carriageway and along position of a road vehicle. Hensel et al. [12] showed a particle filter approach for railways based on just an eddy current sensor and a map.

Only few approaches evaluate data sets with statistics about track selectivity: Lauer and Stein [13] used GNSS and a velocity sensor and showed a gain in track-selective accuracy and confidence between a simple map-match and a proposed estimation algorithm. Hensel et al. [12] showed no direct figures on *switch way* resolution but improved switch detection (98.23%) and classification (99.64%) with an eddy current sensor of 861 switches. This study focused on switches as position input and classifications on merging and splitting switch runs. Böhringer [6] evaluated an integrated navigation system (GNSS, IMU) in combination with an eddy current sensor for switch way identification. Even with a moderate switch detection rate of 70%, the results received 99.78% of track-selective accuracy. These results are considered as most suitable for a comparison and are based on real train runs of 120 km with 113 switches.

## 3. Map-Based Railway Navigation

**3.1. Topological Coordinates.** The goal of train localization is to estimate the train position in the track network by topological coordinates as well as the train speed  $v$ . A unique and discrete track ID ( $id$ ) identifies the track and the track length variable  $s$  is the one-dimensional position on that track. Each track has an origin and a direction  $dir$  indicates if a train is oriented with or against the track definition. The topological coordinates are

$$T^{\text{topo}} = \{id, s, dir\}. \quad (1)$$

Tracks are connected by switches, crossings, or diamond switch crossings. A track  $id$  is defined between connections with a unique ID; that is, it contains no switch or

TABLE 1: Train directions.

Train-track frame direction $dir$	Train velocity direction $m = \text{sign}(v)$	Train-track frame motion direction $\text{sign}(\dot{s})$
+	Forward (+)	+
-	Forward (+)	-
+	Backward (-)	-
-	Backward (-)	+
+	Stop (0)	0
-	Stop (0)	0

crossing. This definition ensures that a track  $id$  is always one-dimensional and limited by the two endings of track beginning and track end.

**3.2. Coordinate Frames.** The sensors measure in their specific *sensor frame*. For further processing, these measurements are converted in the *train frame* according to the mounting parameters. The map and especially the geometry of the tracks are expressed in the *track frame*. Any sample point of a track contains a geographic position (WGS84) and the track attitude angles are defined from a local frame in north, east, and down (NED). Once angles or curvatures are used in the map, there is an ambiguity about the direction at which the track ending the origin is defined. Therefore, it is necessary to define a start point and consequently a pointing direction of the track. A train (train frame) can be placed in two orientations on a track and can move forward or backwards. Alternatively to  $v$ , the train speed can be expressed between train and track frame by  $\dot{s}$ . The absolute velocities of  $v$  and  $\dot{s}$  are the same. The sign of  $\dot{s}$  indicates either an increasing (+) or decreasing (-) change of the position  $s$  of the current track and depends on the track definitions.

Throughout this approach, the map information of a specific location is converted and processed in the *train frame*. It should be noted that there is an alternative way to convert all variables to the *track frame*. Table 1 shows the conversions of the train direction variables: train-to-track frame direction ( $dir$ ), the motion direction of the train ( $m$ ), and the direction of motion between train and track frame by the sign of  $\dot{s}$ . The motion direction  $m$  depends on a forward or backward velocity  $v$  regarding the train frame definition of front and rear. It is possible to compute any of these directions from the two others by Table 1. Direction  $dir$  keeps its value during standstill. The motion direction  $m$  remains the same during a train run, while the sign of  $\dot{s}$  and  $dir$  can alternate after a change of tracks during a train run.

**3.3. Along Track and Cross Track.** A suitable analysis for topological localization performance is the approach by along and cross track. Along track addresses the continuous 1D localization on a track. Cross track focuses on discrete, different tracks and track selectivity is the ability of a correct cross track localization. Sensors can contribute to along and cross localization with relative or absolute measurements of displacement or track features. As shown in Figure 2,

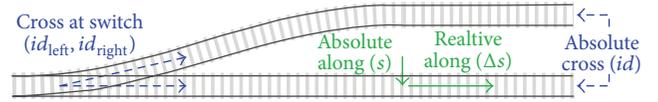


FIGURE 2: Along and cross track definitions.

measurements can contribute to train localization in four different ways:

- (i) relative along ( $\Delta s$ ): odometry,
- (ii) absolute along ( $s$ ): diverse along-track features,
- (iii) cross at switch ( $id$ ): competing switch way track features,
- (iv) absolute cross ( $id$ ): diverse cross/parallel track features.

Odometry is the processing of relative along measurements, such as wheel turns, speed, and train acceleration. Depending on track features, parallel tracks show often very similar along and cross track features. Measurements may contribute to absolute along or cross in a local vicinity or even globally.

**3.4. Railway Track Features.** A suitable track feature is an unchanging property of the track which can be measured by a sensor. Over different locations, there are unique track features as well as repeated, ambiguous features possible. Here, the features are the track geometry by geographic position (latitude  $\varphi$ , longitude  $\lambda$ ), track attitude (i.e., heading  $\psi$ ), and curvature. The heading  $\psi(s)$  changes over the run of the track, which is represented by the heading curvature  $c_\psi = d\psi(s)/ds$ . Additional features can be extended, provided that there is a reproducible signal over different locations and sensors can measure these features.

**3.5. Railway Switch.** The switch way identification is a critical process in railway navigation, especially if the tracks are parallel after the train passes a splitting switch. As a special property of a switch, the two tracks of the competing switch ways differ in geometric characteristics of curvatures  $c^\psi$ , headings  $\psi$ , and geopositions (latitude  $\varphi$ , longitude  $\lambda$ ). The geopositions of left and right switch way are located apart by the cross track distance  $d_{CT}$ . The switch way track positions and headings increase slowly from the switch start, while the curvature is already present from the switch start. There can be many more switch way features, as shown in other approaches based on different sensors [12].

**3.6. Track Map.** The railway map works as a coordinate transformation between topological coordinates and track features (e.g., geometric coordinates). The map model contains and connects information on topology and track features. These track features are parametrized by the 1D-position  $s$  and stored in discrete points (track points). A continuous representation of track features is achieved by interpolations

between these points. The track features can be obtained from the map by the topological pose and converted to train frame:

$$f_{\text{map}}(id, s, dir) = \left\{ \underbrace{\varphi, \lambda}_{\text{geo position}}, \underbrace{\hat{\psi}}_{\text{attitude}}, \underbrace{\hat{c}^\psi}_{\text{curvature}} \right\}^{\text{train}}. \quad (2)$$

According to the train direction  $dir$ , the sign of the curvature  $c^\psi$  changes, while  $\psi$  changes by  $180^\circ$ . This special map can be constructed from train-side sensor data [7, 21] or extracted from an existing geodatabase.

An open street map (OSM) [22] geodatabase is used as data source for the track map. The main advantages of OSM are the availability and the completeness of data points of many tracks of the desired railway track network. For an adequate map, there is some additional preparation by track separation and topological connection (e.g., at switches) as well as geometry processing being necessary. An OSM data contains geodata points  $(\varphi, \lambda)$  and the track geometry of heading  $(\psi)$  and heading curvature  $(c^\psi)$  is derived from these positions. The data is usually collected from various sources, such as a GNSS hand-held or extracted from aerial or satellite photo with different and undefined accuracy. Therefore, an OSM data based map should not be considered as highly accurate. An analysis of many train runs showed lateral deviations up to 5 m between an averaged GNSS trace and the OSM track.

**3.7. Train State Estimation.** The estimation state of railway localization is formulated with the following random variables:

- (i) track ID:  $id$  (discrete),
- (ii) position:  $s$  (continuous, only within a track),
- (iii) train direction on track:  $dir$  (binary),
- (iv) train speed:  $v$  (continuous),
- (v) train acceleration:  $a$  (continuous),
- (vi) correlated sensor properties: biases:  $b$  (continuous).

The estimation state vector or *train state*  $T_k^{\text{est}}$  for one discrete time step  $k$  is defined by

$$T_k^{\text{est}} = \left\{ \underbrace{id, s, dir}_{\text{topological pose}}, \underbrace{v, a}_{\text{train motion}} \right\}_k. \quad (3)$$

The train-to-track frame speed  $\dot{s}$  can be computed by Table 1:

$$\dot{s}_k = \begin{cases} v_k, & \text{if } dir = +, \\ -v_k, & \text{if } dir = -. \end{cases} \quad (4)$$

The bias vector  $B_k$  contains correlated sensor errors. These biases change over time due to a random drift, which cannot be calibrated in advance. Additionally to the state variables, an auxiliary variable  $m$  is used to represent the vehicle motion by  $m \in \{\text{forward, stop, backward}\}$ . The goal for a train localization algorithm is to estimate and resolve  $T_k^{\text{est}}$  and  $B_k$ .

The strong track-train constraint allows to predict the train position, attitude, and inertial state from the known track geometry of the map. These extended train states are computed from the map by the actual topological position estimate and contains the geometry in train frame ( $\psi$  and  $c^\psi$  transformation according to  $dir$ ):

$$T_k^{\text{ext}} = \{\varphi, \lambda, \hat{\psi}, \hat{c}^\psi\}_k^{\text{train}}. \quad (5)$$

**3.8. Train Control.** The train control consists of cross track control and the along-track control by the train driver:

$$U = \left\{ \underbrace{U^{\text{sw}}}_{\substack{\text{cross control,} \\ \text{control center}}}, \underbrace{U^{\text{acc}}, U^m}_{\substack{\text{along control,} \\ \text{train driver}}} \right\}. \quad (6)$$

The cross control influences the travel path of the train by the selected switch way ( $U^{\text{sw}} : \{\text{left, right}\}$ ). This is usually controlled by a train control center or sometimes by the train driver at shunting yards or industrial tracks. The train driver controls the general train motion  $U^m$  and the acceleration  $U^{\text{acc}}$  by the traction and brake lever. The general train motion is the travel direction selector as well as a train stop (e.g., activated parking brakes). The control center has also influence on the along-track control via signaling.

**3.9. Simple Map-Matching.** In contrast to state estimation a reference approach by simple map-matching is described. The simple map-matching is a snapshot based, nearest neighbor method, which uses no information about the prior position. The nearest position on track is computed with the map from a GNSS position measurement:

$$\underbrace{\{id, s\}}_{\text{topo position}} = f_{\text{map-match}} \left( \underbrace{\varphi, \lambda}_{\text{geo position}} \right). \quad (7)$$

It should be noted that this approach would be sufficient if the position measurement (e.g., ideal GNSS) is continuously available with an accuracy always better than half of the distance of parallel tracks. It will be shown in the results that this approach has its problems with real GNSS position measurements.

## 4. Train-Side Sensor Measurements

This paper focuses only on GNSS and IMU measurements, but other train-born sensors can extend the proposed approach. The used sensors are considered as inexpensive and their combination as complementary in terms of measurement errors.

**4.1. Global Navigation Satellite Systems.** The approach uses the standard GNSS receiver output by position-velocity-time (PVT) and is considered as loosely coupled. That means that the internal GNSS related computations can be kept transparent, up-to-date with actual receiver technology and out of the train localization approach. The used receiver was

a u-blox 6 GPS receiver, but also other commercial off-the-shelf receivers could be used.

A drawback of GNSS is the lack of availability and degraded accuracy in parts of the railway environment. GNSS data is not available in tunnels or below station roofs. The accuracy is further affected by multipath, signal loss, and poor geometry in urban environments, next to acoustic noise barriers or in dense forests.

**4.1.1. GNSS Speed.** The GNSS speed measurement contributes to the train odometry in terms of *relative along-track* estimation and is part of the PVT data. A single antenna mounting is invariant of a horizontal rotation; that is, the speed and also heading measurements are independent of the mounted yaw angle between antenna and train. In principle, the GNSS speed is computed from the vector norm of the antenna motion in north and east component. Therefore, the GNSS speed is always positive and the train motion ( $m$ ) must be considered in the use-case for train speed. The measurement model contains the train speed  $\dot{s}$ , additional white noise  $n$ , and a conversion for the train motion:

$$Z_k^{\text{GNSS},v} = \begin{cases} -v_k + n_k^v, & \text{if } m = \text{backwards}, \\ v_k + n_k^v, & \text{if } m = \text{forwards}. \end{cases} \quad (8)$$

**4.1.2. GNSS Positions.** GNSS positions are a favorable measurement for absolute along and cross resolution of the train location. The GNSS position model includes 2D positions of latitude  $\varphi$  and longitude  $\lambda$  and additional white noise  $n^{\text{pos}}$ :

$$Z_k^{\text{GNSS, pos}} = \begin{pmatrix} \varphi \\ \lambda \end{pmatrix}_k + n_k^{\text{pos}}. \quad (9)$$

**4.1.3. GNSS Heading.** The GNSS heading angle contributes to the switch resolution which was described as *cross at switch* contribution in Section 3.3. There are multiple ways of the internal heading computation of the receiver: a simple differentiation of two consecutive positions or by computation of a receiver motion vector from Doppler and/or phase measurements of each satellite. The actual internal method is unknown, but state-of-the-art methods use positions and Doppler within a Kalman filter [23]. The used receiver outputs an ECEF (earth centered, earth fixed) velocity vector, and the GNSS heading measurement is the angle to north of the horizontal part of this vector in east and north direction. The receiver estimates further an accuracy of the velocity vector (ECEF) better than 1.1 km/h in 95% of all runs and a resulting heading accuracy of 1.7° in 95% of all runs above 10 km/h. The GNSS heading is worse at low speeds, so the heading measurements are used above 10 km/h. The GPS data of the experiment and the results showed a good heading repeatability with usually less than 1° of different runs over the same positions. In comparison, a heading angle between two consecutive positions showed much worse results.

The heading model contains the heading angle  $\psi$  of the train (train frame) which is derived from the estimate of the topological pose and the map. A GNSS heading measurement

with single antenna is the heading of the antenna motion and requires a conversion to train frame by the train motion  $m$  (see also Table 1):

$$Z_k^{\text{GNSS},\psi} = \begin{cases} \hat{\psi}_k + \pi + n_k^\psi, & \text{if } v < 0 \text{ (} m = \text{backwards)}, \\ \hat{\psi}_k + n_k^\psi, & \text{if forwards.} \end{cases} \quad (10)$$

It should be noted that there are no heading measurements for very low speeds or stopped trains.

**4.2. Inertial Sensor.** The IMU measurements provide continuous and interference-free data. Two measurements are of particular interest: the longitudinal acceleration ( $a^x$ ) and the yaw turn rate ( $\omega^z$ ). The measurements are aligned with the train frame by prior calibration.

**4.2.1. Longitudinal Acceleration.** The longitudinal acceleration ( $a^x$ ) measures the train acceleration measurement and it contributes to the relative along localization (odometry). The along acceleration measurement model considers train acceleration  $a$ , the gravity  $g$  portion by the slope angle  $\theta$  of the track, and accelerometer bias  $b^{ax}$ :

$$Z_k^{\text{IMU},ax} = a_k + \frac{g \cdot \sin \theta_k + b_k^{ax}}{\tilde{b}_k} + n_k^{ax}. \quad (11)$$

Inertial sensors are affected by a bias which is changing over time, called drift. Any bias causes growing errors, as the train acceleration is integrated over time to speed and position. The presented approach does not contain a slope estimation or slope profile in the map. The slope angles of railway tracks are relatively small and the bias estimation in the following handles the difference. A continuous estimation of the combined offset  $\tilde{b}_k$  is processed and contains bias and gravity portion from slope. Extensions with slope estimation by integration or a slope profile in the map are possible.

**4.2.2. Yaw Turn Rate.** Rails are a strong constraint between track geometry and train trajectory. It is possible to measure the geometric characteristics with a train-side IMU of a moving train. A certain curvature  $c^y$  of a track causes yaw turn rates depending on the train speed:

$$\hat{c}^y = \frac{\dot{\psi}}{v}. \quad (12)$$

A complete model of train kinematics for turn rates and centripetal accelerations is presented in [21]. Trains are exposed to low pitch and roll angles in general and especially at switches. As a consequence, the horizontal heading turn rate  $\dot{\psi}$  can be approximated by the yaw rate of the train:

$$\dot{\psi} \approx \omega^z. \quad (13)$$

The curvature as measurable track feature indicates absolute along locations and resolves different switch ways (cross at

switch). This approach is less sensitive to drift as it does not rely on integration of the inertial measurements. This property is advantageous for the use of low-cost MEMS gyroscopes. The measurement model  $h_\omega$  of the yaw rate measurement  $\omega_z$  is defined by

$$Z_k^{\text{IMU},\omega z} = \frac{\tilde{c}_k^y \cdot v_k + b_k^{gz}}{h_{\omega z}(T_k, B_k)} + n_k^{\omega z}. \quad (14)$$

At this point, the frame and motion definitions are vital (Table 1). The curvature is translated to train frame by train direction  $dir$  and the speed contains the motion direction  $m$  in its sign. The bias of the gyroscope ( $b^{gz}$ ) is calibrated only during stop phases of the train:

$$b_k^{gz} \approx \omega^z, \quad \text{if } m = \text{stop}. \quad (15)$$

The turn rates are assumed to be zero and the low earth turn rate is neglected. The resulting small error is not integrated in the following (e.g., as in strapdown approaches) and results in a negligible error at the weighting process by a likelihood.

## 5. Probabilistic Train Localization

The following definitions can be used for a multihypothesis filter, a particle filter or Rao-Blackwellized particle filter. The posterior is already specified for GNSS and IMU sensors.

**5.1. Train Localization Posterior.** The train localization posterior represents the estimation problem. In previous works [3, 4, 21], Bayesian methods are presented with a dynamic Bayesian network definition for the train localization problem and the factorization of the posterior is shown in steps. The posterior of all train states ( $T_{0:k}$ ) and sensor biases ( $B_{0:k}$ ) over  $k$  time steps are estimated given all measurements ( $Z_{0:k}$ ), train control inputs ( $U_{0:k}$ ), and the map ( $M$ ). The map is actually known in advance and does not change over time for the train localization problem. It is included in the conditional part of the distributions to indicate where information of the map is needed. The unknown train control is also included to indicate where a train driver or train control can influence the train states. The posterior is factorized in a recursive form in order to compute the posterior practically with estimation algorithms:

$$\begin{aligned} & p(T_{0:k}, B_{0:k} | Z_{1:k}, U_{0:k}, M) \\ & \propto \underbrace{p(Z_k^{\text{GNSS}} | T_k) \cdot p(Z_k^{\text{IMU}} | T_k, B_k)}_{\text{measurements}} \\ & \quad \cdot \underbrace{p(T_k | T_{k-1}, U_k, M)}_{\text{train transition}} \cdot \underbrace{p(B_k | B_{k-1})}_{\text{bias transition}} \\ & \quad \cdot \underbrace{p(T_{0:k-1}, B_{0:k-1} | Z_{1:k-1}, U_{1:k-1}, M)}_{\text{recursion}}. \end{aligned} \quad (16)$$

The factorized posterior is proportional ( $\propto$ ) to the posterior, because of a missing normalization factor. This normalization can be computed separately, as all probabilities sum up to one.

There are different considerable filter approaches, which are able to estimate the nonlinear train transition and the different hypotheses of the topological pose.

**5.2. Along Track.** One extension to the previous particle filter approaches [3, 4] is the separation of states, where certain states are estimated differently. In combination with particle filters, this process is called Rao-Blackwellization [24]. In particular the linear states can be separated from the nonlinear states of a particle hypotheses and estimated by a more optimal filter, as the Kalman filter. The train state contains linear and nonlinear parts:  $T_k = \{T_k^l, T_k^n\}$ . The train transition of 16 is split in a linear and nonlinear part via chain rule and conditional independencies that are removed:

$$\begin{aligned} p(T_k | T_{k-1}, U_k, M) &= p(T_k^n, T_k^l | T_{k-1}^n, T_{k-1}^l, U_k, M) \\ &= \underbrace{p(T_k^n | T_{k-1}^n, U_k, M)}_{\text{nonlinear train transition on tracks}} \\ & \quad \cdot \underbrace{p(T_k^l | T_{k-1}^l, U_k)}_{\text{linear train transition}}. \end{aligned} \quad (17)$$

**5.2.1. Train Odometry Filter.** The linear and one-dimensional train transition is separated from the nonlinear transition on the map and estimated with a Kalman filter. The estimation of the linear state variables of acceleration  $a$ , speed  $v$ , and displacement  $\Delta s$  will be called *odometry* in the following. The odometry combines also the along acceleration bias estimation with updates from acceleration and speed measurements:

$$\begin{aligned} & \underbrace{p(T_k^l | T_{k-1}^l, U_k) \cdot p(B_k^{\text{ax}} | B_{k-1}^{\text{ax}})}_{\text{odometry prediction}} \\ & \quad \cdot \underbrace{p(Z_k^{\text{GNSS},v} | T_k^l) \cdot p(Z_k^{\text{IMU},\text{ax}} | T_k^l, B_k^{\text{ax}})}_{\text{odometry update}}. \end{aligned} \quad (18)$$

The state transition of the odometry prediction step is defined here as a 1D transition DWPA model (discrete white noise constant acceleration) [25]. The linear 1D train transition and the acceleration bias are estimated by the discrete model:

$$\begin{pmatrix} \Delta s \\ v \\ a \\ \tilde{b} \end{pmatrix}_k = \underbrace{\begin{pmatrix} 0 & \Delta t & \frac{\Delta t^2}{2} & 0 \\ 0 & 1 & \Delta t & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{system matrix}} \begin{pmatrix} \Delta s \\ v \\ a \\ \tilde{b} \end{pmatrix}_{k-1}. \quad (19)$$

This model is propagated by state-of-the-art Kalman filter and updated with GNSS speed and longitudinal IMU acceleration measurements.

**5.2.2. Track Transition Model.** The track transition ensures that the estimates (hypotheses, particles) exist and stay exclusively on tracks. The nonlinear train transition of (17)

estimates a topological coordinate from the linear displacement, the previous coordinate, an unknown switch way, and the map:

$$\begin{aligned} p(T_k^n | T_k^i, T_{k-1}^n, U_k, M) \\ = p(T_k^{\text{topo}} | T_{k-1}^{\text{topo}}, \Delta s_k, U_k^{\text{sw}}, M). \end{aligned} \quad (20)$$

The function of the map for each hypothesis (particle) is

$$T_k^{\text{topo}} = f_{\text{map,trans.}}(T_{k-1}^{\text{topo}}, \Delta s_k). \quad (21)$$

This model considers the discontinuity at a track change and in case of a splitting switch scenario the next track is sampled from a discrete uniform distribution (left or right). According to the motion state of standstill ( $m = \text{STOP}$ ), this transition can be suspended. The next step is a track map query by the topological pose for each hypotheses or particle. The extended train state  $T_k^{\text{ext}}$  contains the track geometry in train frame:

$$T_k^{\text{ext}} = f_{\text{map,data}}(T_k^{\text{topo}}). \quad (22)$$

**5.3. Cross Track.** The cross track estimation evaluates different tracks at a switch or in scenarios with multiple tracks. The evaluation or weighting process is based on a soft comparison of measurements and expected measurements from the map. A sensor likelihood function is defined for each measurement type. First, the expected or estimated measurement is computed from a measurement model and the current state of train state and sensor correlation  $Y_k = h(T_k, B_k)$ . A generic likelihood function model is defined here with a Gaussian distribution. The mean is the expected measurement  $Y_k$ , the argument is  $Z_k$ , and the covariance of the sensor noise is  $\Sigma$ :

$$\begin{aligned} p(Z_k | T_k, B_k) \\ \cong |2\pi\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(Z_k - Y_k)\Sigma^{-1}(Z_k - Y_k)^T\right\}. \end{aligned} \quad (23)$$

## 6. Particle Filter Approach

A particle filter is chosen for the posterior (16) estimation. The particle filter can handle different nonlinear estimates (hypotheses) automatically by the particles, which is necessary for a distribution of any possible position over different tracks. As described in [26], a particle filter represents probability density functions by appropriate particle distributions with appropriate weights of  $N_p$  particles. The posterior of (16) is represented by the particles set:

$$p(\{T, B\}_{0:k} | Z_{1:k}, U_{0:k}, M) \approx \{x_{0:k}^i, w^i\}_{i=1}^{N_p}. \quad (24)$$

$x_{0:k}^i$  is the  $i$ th particle with its weight  $w^i$  of  $N_p$  particles and represents one sample of the posterior of all time steps until  $k$ . Particles are generated from a function which is easy to calculate [26], called the proposal function:

$$x_{0:k}^i \sim q(\{T, B\}_{0:k} | Z_{1:k}, U_{0:k}, M). \quad (25)$$

Afterward these particles are weighted [26]. The weights are proportional to the fraction posterior over proposal function:

$$w_k \propto \frac{p(\{T, B\}_{0:k} | Z_{1:k}, U_{0:k}, M)}{q(\{T, B\}_{0:k} | Z_{1:k}, U_{0:k}, M)} \cdot \frac{1}{\bar{w}, \text{unnormalized weights}}. \quad (26)$$

The weights of the particle filter sum up to one and the normalization factor  $\eta$  can be easily computed:

$$\eta = \frac{1}{\sum_{i=1}^{N_p} w_k^i}. \quad (27)$$

After several time steps, some particles may carry an extreme high weight while the rest has a very low weight. These low weighted particles are inefficient and this process is called degeneration. In order to avoid this, a systematic resampling [26] of the particle distribution can solve this problem. A metric for particle depletion is the effective number of particles  $\widehat{N}_{\text{eff}}$  [25]. The particle distribution is resampled if  $\widehat{N}_{\text{eff}}$  is below a threshold  $N_{\text{th}}$ .

### 6.1. Particle Filter with GNSS and IMU Measurements

**6.1.1. Proposal Function.** The proposal function is designed by the transitions of train and correlated sensor properties as well as suitable measurements. The proposal contains the train odometry estimation of Section 5.2.1 and a gyroscope bias estimation. In this implementation, the gyroscope bias is only updated if the train is not moving. The displacement of the  $i$ th particle is sampled from the odometry Kalman filter output:

$$\Delta s_k^i \sim \mathcal{N}(\Delta s_{\text{KF}}^i, \sigma_{\Delta s}^2). \quad (28)$$

In contrast to previous approaches [3, 4], the samples were directly generated from an acceleration distribution and a motion model. The use of the odometry Kalman filter inside of a particle filter is the Rao-Blackwellization part. The nonlinear map transition of (21) and map query (22) is processed for each particle as a function of the map.

**6.1.2. Weight Function.** The weight function is the combination of IMU and GPS likelihoods and computed as shown in (23) with the appropriate measurement models (9), (10), and (14). The weight function for the  $i$ th particle with the IMU and GNSS likelihoods is

$$\begin{aligned} w_k^i = \eta \cdot w_{k-1}^i \cdot p(Z_k^{\text{GNSS,pos}} | T_k^i) \cdot p(Z_k^{\text{GNSS},\psi} | T_k^i) \\ \cdot p(Z_k^{\text{IMU},\omega_z} | T_k^i, B_k^i). \end{aligned} \quad (29)$$

**6.1.3. Initialization.** The initial particle distribution is generated from the first GNSS position measurement. Therefore, the 2D geoposition is sampled  $N_p$  times with a large covariance  $\Sigma$  (e.g., 100 m) from a Gaussian distribution:

$$\overline{\text{pos}}^i \sim \mathcal{N}(\text{pos}_{k=1}^{\text{GNSS}}, \Sigma). \quad (30)$$

The positions are map-matched by (7) to topological positions and assigned to each particle  $i$ . The train frame direction  $dir$  is sampled randomly from a uniform distribution.

**6.2. Output Estimate.** A particle distribution is a less useful output for applications like automated train control or collision avoidance. There, a single mode or most likely output is desired. Internally, the particle filter keeps its particle distribution for the next update. The output is computed by four steps: first, the track paths are identified and, second, the most likely track path with particles is chosen (ML path). A mean square estimate is computed from particles on that path and the result is translated back to topological coordinates.

**6.2.1. Track Path Identification.** At first, all path hypotheses  $id_p$  with at least one particle are identified:

$$id_p^{0:j} = f_{\text{findPaths}} \left( \{id^i\}_{i=1}^{Np} \right). \quad (31)$$

A track path contains one or more sequential tracks ( $id$ ) on a 1D path, where a train is able to run over in a sequence. For further computations, a track path has a continuous 1D coordinate frame compared to discontinuities at the joints of tracks. A topological pose can be translated into path coordinates as well as translated from path coordinates. As an example, in case of a split switch scenario, particles can be distributed before the switch, on the left and right switch way. This would result in two possible track paths:  $id_p^1 = \{id_{\text{before}}, id_{\text{left}}\}$  and  $id_p^2 = \{id_{\text{before}}, id_{\text{right}}\}$ .

**6.2.2. ML Path.** As a next step, the sum of weights are calculated for each path  $j$ :

$$\widehat{w}^j = \sum_{i=1}^{Np} w^i \cdot \underbrace{\delta(id^i = id \in id_p^j)}_{\substack{\text{selects weight of } i\text{th particle} \\ \text{from the } j\text{th path } id_p}}. \quad (32)$$

The delta function  $\delta(id^i = id \in id_p^j)$  equals one if the  $i$ th particle (respective its track ID  $id$ ) is on the  $j$ th track path and is zero otherwise. The most likely path  $id_p^{\text{ML}}$  is the path index  $J$  with the highest cumulative weight  $\widehat{w}^{\text{ML}}$ :

$$J = f_{\text{arg max } j} \left( \widehat{w}^j \right). \quad (33)$$

**6.2.3. Mean Square Estimate on Path.** In the following, a delta function selects the  $i$ th particle which is on the most likely path:

$$\delta^i = \delta \left( id^i = id \in id_p^{\text{ML}} \right). \quad (34)$$

The topological coordinates of the selected particles are translated to the most likely path:

$$\widetilde{T}^{\text{topo},i} : \{id, \widetilde{s}, \widetilde{dir}\} = f_{\text{topo2path}} \left( T^{\text{topo},i}, id_p \right). \quad (35)$$

The 1D position is calculated by a weighted mean of the selected particles, which belong to the ML path:

$$\widehat{s} = \frac{1}{\widehat{w}^{\text{ML}}} \sum_{i=1}^{Np} (\widetilde{s}^i \cdot w^i \cdot \delta^i). \quad (36)$$

The 1D position deviation (along-track precision) is calculated by the weighted sample variance of particles from  $id_p^{\text{ML}}$ :

$$\widehat{\sigma}_s = \sqrt{\frac{1}{\widehat{w}^{\text{ML}}} \sum_{i=1}^{Np} \left( (\widetilde{s}^i - \widehat{s})^2 \cdot w^i \cdot \delta^i \right)}. \quad (37)$$

The train direction on track  $dir$  is computed by the highest weight, where  $l$  is either positive or negative track frame direction of the path:

$$\begin{aligned} \widehat{w}^{\text{ML},+} &= \frac{1}{\widehat{w}^{\text{ML}}} \sum_{i=1}^{Np} \left( \delta \left( \widetilde{dir}^i = + \right) \cdot w^i \cdot \delta^i \right), \\ \widehat{w}^{\text{ML},-} &= \frac{1}{\widehat{w}^{\text{ML}}} \sum_{i=1}^{Np} \left( \delta \left( \widetilde{dir}^i = - \right) \cdot w^i \cdot \delta^i \right). \end{aligned} \quad (38)$$

The most likely direction is the one with the higher weight. The procedure for train motion direction  $m$  is analog.

**6.2.4. Translation to Topological Pose.** Finally, the most likely path  $id_p^{\text{ML}}$ , the weighted mean position  $\widehat{s}$  on path, and the train to path frame direction  $\widetilde{dir}_{\text{ML}}$  are translated into topological coordinate frame:

$$T^{\text{topo}} : \{id, s, dir\} = f_{\text{path2topo}} \left( id_p^{\text{ML}}, \widehat{s}, \widetilde{dir}_{\text{ML}}^{\text{ML}} \right). \quad (39)$$

**6.3. Algorithm Summary.** The algorithm of the sequential Bayesian filter with a Rao-Blackwellized filter realization (RBPF) is shown in Algorithm 1 and summarized in words: every new measurement (IMU or GNSS) triggers the filter to compute a next time step. Particles (hypotheses) estimate a topological position on railway tracks and these particles exist only on tracks. Each particle is shifted along the track by a displacement output from the odometry filter (Kalman filter), which is updated with GNSS speed and longitudinal acceleration data (IMU). A special function of the map processes this shift for topological coordinates. A railway switch is handled by a random assignment of the particles to each way. From the map, each particle is assigned with proposed geometric values in train frame of a geoposition, heading angle, and an instantaneous turn rate from curvature and speed. The likelihoods weigh the particles with a probability according to the difference of the proposed geometric values and the measurements (train frame) of GNSS position, GNSS motion vector heading, and IMU yaw rate. An output estimate extracts one train location with variances from the particles and resampling removes unlikely particle estimates.

The design parameters are sensor (co)variances, process noise of train acceleration, and combined bias as well as number of particles, resample threshold, and variance of displacement sampling.

**6.4. Particle Filter Challenges.** The challenges of this map-based localization approach by a particle filter are the following.

```

Algorithm: Train Localization (RBPF)
Input: GNSS and IMU sensor data
Output: topological coord. ( $id, s, dir$ ) and train speed
(1) load map
(2) initialize odometry Kalman filters with zero vector
(3) initialize all  $N_p$  particles by first GNSS position (30)
(4) loop
(5)   if new measurement(s) available then
(6)     time step:  $k = k + 1, \Delta t = t_k - t_{k-1}$ 
(7)     for all  $N_p$  particles do
(8)       predict odometry KF (19)
(9)       update KF with speed (8)/acceleration (11)
(10)      if train is moving then
(11)        sample displacement from odometry (28)
(12)        compute map transition (21)
(13)        get geometry from map (train frame) (22)
(14)        compute likelihoods (9)/(10)/(14)
(15)        multiply particle weight by likelihoods (29)
(16)      else (train is stopped)
(17)        observe and filter gyroscope bias
(18)      end if
(19)    end for
(20)    normalize weights (27)
(21)    compute most likely output estimate (31)–(39)
(22)    if resampling necessary by  $N_{\text{eff}}$  then
(23)      perform resampling
(24)    end if
(25)  end if
(26) end loop

```

ALGORITHM 1: Algorithm of the map-based train localization with GNSS, IMU, and Rao-Blackwellized particle filter.

**6.4.1. Divergence in Along Track.** The problem is an unstable filter, as the estimate (particles) is away from the truth and cannot recover. This can be approached by a good model design (i.e., proposal function), a continued resampling, and the insertion of sampling noise. In case it fails, a filter monitoring can detect a severe along divergence (e.g., by GNSS measurements) and restart the filter.

**6.4.2. Divergence in Cross Track.** The problem here is a failed track selectivity, as all particles are on the wrong track, in particular a parallel track, and cannot recover. The resolution of the switch way is very important. This requires a sufficient map model as well as sensors, which are able to measure the competing switch way properties. Another approach is the use of extrinsic sensors which directly measure the switch way (cross at switch) or observe a neighboring track (absolute cross).

**6.4.3. Initialization.** In the start-up phase of the estimation filter, some hidden and discrete states remain unresolved if they are not directly observable by measurements. In the proposed approach, this happens for the train frame direction  $dir$  and the track  $id$  in parallel track scenarios. There, the filter requires motion or motion over a switch to resolve these states. An alternative way is the use of extrinsic sensors

in order to observe or resolve the hidden variables from standstill.

**6.4.4. Overconfidence.** This happens especially if the measurement noise is too small and correlations are disregarded in the sensor model. The filter converges very quickly to the measurements and results in a too small particle distribution after resampling. This can lead to the described divergence in along and cross track.

**6.4.5. Degeneracy.** Degeneracy of the particle distribution is the effect where nearly all weight is accumulated on one or a few particles. The state-of-the-art approach is systematic resampling [26].

**6.4.6. Dimensionality and Computational Complexity.** High dimensional state vectors can be problematic for particle filters, as the number of particles and computational complexity grows [27]. This approach uses a state vector (3) with the nonlinear random variables of discrete track IDs  $id$ , which is limited to a few tracks in the vicinity, a noncontinuous ID position  $s$ , and a binary direction  $dir$ . Once, the direction is resolved after initialization, the direction is processed in a deterministic manner and not estimated anymore. In other words, the particle filter estimates actually two states.

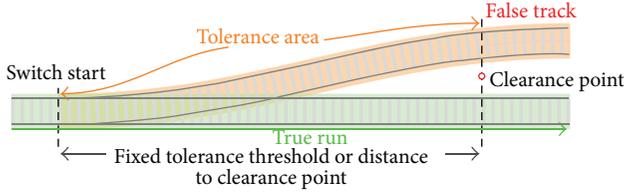


FIGURE 3: Cross track analysis at a switch with tolerance and error areas for a true right (straight) run.

The train acceleration, speed, and certain biases are processed by a linear filter in order to achieve more optimal estimation and to reduce the number of particles.

## 7. Track-Selective Evaluation Framework

The empirical proof of track selectivity is achieved by the comparison of localization output and a reference route. This reference must be known in advance and is the true sequence of traveled track IDs. Of special interest is the switch scenario with a splitting switch way run, as shown in Figure 3. Railway switches have a region where the clearance of two vehicles overlap, and only one train can occupy these tracks. There, a false track estimate is tolerated and not a real problem as only one train can occupy the tracks. The length of this tolerance region can either be fixed or individual for every switch, stored within the map. For simplicity reasons, a fixed tolerance of 50 m after a switch start is chosen. The track-selective *accuracy* is evaluated with the known route (ground truth) and a false track estimate within the tolerance region is marked in orange, a correct estimate is green, and a false one is red. Track *precision* is defined here as the discrete probability of the track estimate from the filter output. A high precision estimate can be evaluated with an incorrect accuracy, when the true track is different.

The track-selective accuracy can be analyzed over time (per second) or over traveled distance (per meter). Train statistics are often related to distance (e.g., millions of train kilometers), so the results are presented in relation to the traveled distance. The method of Algorithm 2 evaluates the train localization estimate of each time step.

A cumulative evaluation shows the performance of the localization approach in terms of track selectivity for larger data sets. Each evaluation result is shown relative to the total distance:

$$\begin{aligned}
 E_{\text{error}}^{\text{cum}} &= \frac{\sum e_k^{\text{error}}}{\sum \Delta s_k} \cdot 100\%, \\
 E_{\text{switch}}^{\text{cum}} &= \frac{\sum e_k^{\text{switch}}}{\sum \Delta s_k} \cdot 100\%, \\
 E_{\text{OK}}^{\text{cum}} &= 100\% - E_{\text{switch}} - E_{\text{error}}.
 \end{aligned} \tag{40}$$

This method is based on distances which automatically rejects the evaluation of stopped and parked trains. Track-selective errors occur in the presence of parallel tracks. A train run on a route with more single track scenarios will distort an

*Algorithm:* Track Selective Evaluation

**Input:** Train state  $T_{0:k}$ , true track IDs  $A$ , map

**Output:** Evaluation: cumulative  $E^{\text{cum}}$ ,  $E^{\text{cum},P}$

- (1)  $B$ : all track IDs of wrong switch ways from  $A$ , map
- (2)  $P$ : switch positions from  $A$ , map
- (3)  $l_{\text{clear}}$ : clearance length of each switch
- (4) **for all** train states  $T_{0:k}$  **do**
- (5)  $e_k^{\text{switch}} = e_k^{\text{error}} = \Delta p_k = 0$
- (6) **if** track  $id$  is not in  $A$  (true track ID list) **then**
- (7) **if**  $id$  is in  $B$  & distance to switch  $< l_{\text{clear}}$ . **then**
- (8)  $e_k^{\text{switch}} = \Delta s_k$
- (9) **else**
- (10)  $e_k^{\text{error}} = \Delta s_k$
- (11) **end if**
- (12) **end if**
- (13) **if** other track in the vicinity of  $id_k$ ,  $s_k$  (20 m) **then**
- (14)  $\Delta p_k = \Delta s_k$
- (15) **end if**
- (16) **end for**
- (17) compute cumulative evaluation  $E^{\text{cum}}$  (40)
- (18) compute cum. eval. of parallel tracks  $E^{\text{cum},P}$  (41)
- (19) **return**  $E^{\text{cum}}$ ,  $E^{\text{cum},P}$

ALGORITHM 2: Track-selective evaluation over distances.

evaluation in favor of a better track-selective evaluation. An increase of comparability of the evaluation result is considered with a ratio to distances with parallel tracks in vicinity:

$$\begin{aligned}
 E_{\text{error}}^{\text{cum},P} &= \frac{\sum e_k^{\text{error}}}{\sum \Delta p_k} \cdot 100\%, \\
 E_{\text{switch}}^{\text{cum},P} &= \frac{\sum e_k^{\text{switch}}}{\sum \Delta p_k} \cdot 100\%, \\
 E_{\text{OK}}^{\text{cum},P} &= 100\% - E_{\text{switch}}^P - E_{\text{error}}^P.
 \end{aligned} \tag{41}$$

The switch tolerance evaluation suits mainly for detailed evaluation on small changes and tuning.

A compact figure of the evaluation ( $E$ ) in terms of track selectivity over parallel tracks distances (TS,  $P$ ) is the error-free case of multiple track scenarios:

$$E_{\text{TS},P} = 100\% - E_{\text{error}}^{\text{cum},P}. \tag{42}$$

This figure explains how good a certain train localization approach performs on a specific track network. This cumulative evaluation is one way to measure the track selectivity performance and contains to some extent the track layout of parallel tracks and switch densities.

Another evaluation measure is the error events, which counts and evaluates the transition to the wrong track. One faulty switch resolution results in a cumulative error dependent on the specific track length. A parallel track scenario merges very often by a switch after a station and a wrong output can be on the correct track again. The error event method counts transitions to the error case and respects more the error cause, which is a fault switch way. It differs

TABLE 2: Train routes.

Run	From station	To station	Forward, backward	Split switches	Time	km
1	ABG	FDB	B	8	10 min	8
2	FDB	ABG	F	7	10 min	8
3	ABG	ING	B	22	1 h	66
4	ING	ABG	F	20	1 h	66
5	ABG	FDB	B	8	10 min	8
6	FDB	ABG	F	7	10 min	8
7	ABG	AIC	B	11	30 min	25
8	AIC	ABG	F	9	30 min	25
9	ABG	FDB	B	8	10 min	8
10	FDB	ABG	F	7	10 min	8
$5 \times F, 5 \times B$				107	4 h	230

also in late switch way resolution, if the output is correct again within the correct track which connects to the switch. From these error events, the correct switch way evaluation ( $E_{SW}$ ) statistics can be computed of the total split switches  $N_{total}$ , the late resolved switch ways after the switch tolerance  $N_{late}$  and the wrong resolved switch ways  $N_{failed}$ :

$$E_{SW} = \frac{N_{total} - N_{late} - N_{failed}}{N_{total}} \cdot 100\%. \quad (43)$$

The track-selective evaluation in multiple tracks scenarios  $E_{TS}$  and the switch way resolution evaluation  $E_{SW}$  will be used as compact results.

## 8. Experiment

**8.1. Recorded Data Set.** The data set was recorded on the regional train “Alstom Coradia Lint 41” under regular passenger service conditions. This train can travel up to 120 km/h and has two drivers’ cabs for two-side operation. Table 2 shows the train runs over 230 km with 107 splitting switches. The train runs on 58.5 km of tracks with other tracks near or in parallel, which are 25.4% of all tracks.

The data set contains GNSS PVT data (position, velocity, time) of GPS (Global Positioning System) from a u-blox LEA 6T receiver. The IMU data (Xsens MTi) was recorded with a sample rate of 200 Hz and time stamped from a GPS synchronous clock. For the proposed algorithm, this IMU data was low-pass filtered and downsampled to 4 Hz. The IMU was placed on the front bogie, the GNSS antenna below a fiberglass roof above the bogie position. A special camera (dash-cam) with GPS timestamped video was installed behind the front windshield for the switch way evaluation.

**8.2. Labeled Reference Route.** The reference for the cross track analysis is a recorded video from the train run. In that video, the motion state can be seen, the switch way and direction of travel. For every run, a reference travel path (i.e., labeled data) can be computed from a start position (GNSS), train direction, the map, and a series of true splitting switch ways. These switch ways are either “left” or “right” and obtained manually from the video.

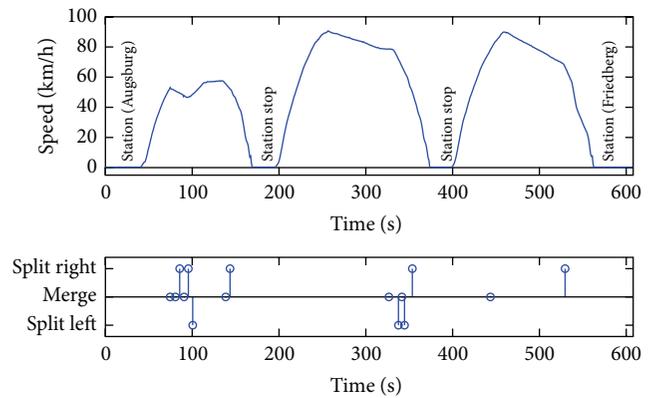


FIGURE 4: Run 1 over time from Augsburg main station to Friedberg station with known switch ways.

**8.3. Implementation.** The localization algorithm approaches were implemented within a self-written JAVA framework. This includes the map processing, the sensor data reader, and the evaluation. The sensor data was processed in a causal way; that is, the localization approach processed each measurement in the chronological order and the output was evaluated. No simulated (i.e., generated) data was used. Particle filters are generally computationally expensive by nature. Nevertheless, the temporal performance for ( $N_p = 100$ ) particles with visualization was processed 19.0 times faster than real time on a laptop (Intel i7M CPU, 2.9 GHz, Windows 7). Hence, a real time operation is possible.

## 9. Results

**9.1. Track Selectivity over Time.** Two different evaluations of the reference approach and the proposed algorithm are shown of Run 1 over time. Therefore, the train speed and true switch ways of Run 1 are shown over time in Figure 4. It visualizes the occurrence of splitting switch ways, since their resolution is the challenge for a train localization filter.

Figure 5 shows the results of the simple map-matched GNSS positions for Run 1 over time. The correct track is marked with OK in green, an error in the tolerance area in

TABLE 3: Detailed cumulative results.

Localization	OK	Switch % total distance: $E^{\text{cum}}$ (% parallel tracks: $E^{\text{cum},P}$ )	Error	Error events, switch way resolutions
(1) Map match GNSS position	93.7 (75.2)	0.73 (2.86)	5.57 ( <b>21.9</b> )	108 errors
(2) RBPF, GNSS position	94.9 (79.8)	1.06 (4.19)	4.06 ( <b>16.0</b> )	33 errors
(3) Method 2 and GNSS heading	97.3 (89.1)	0.26 (1.03)	2.49 ( <b>9.84</b> )	4 switches (1 late, 3 fail)
(4) Method 2 and IMU yaw rate	99.7 (98.9)	0.10 (0.41)	0.17 ( <b>0.67</b> )	3 switches (2 late, 1 fail)
(5) Method 2 and heading, yaw rate	99.7 (98.8)	0.09 (0.36)	0.21 ( <b>0.84</b> )	4 switches (3 late, 1 fail)

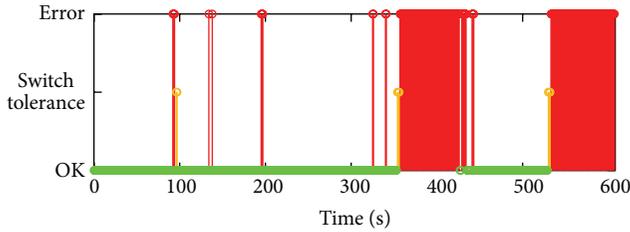


FIGURE 5: Track-selective accuracy of simple map-matching (nearest neighbor).

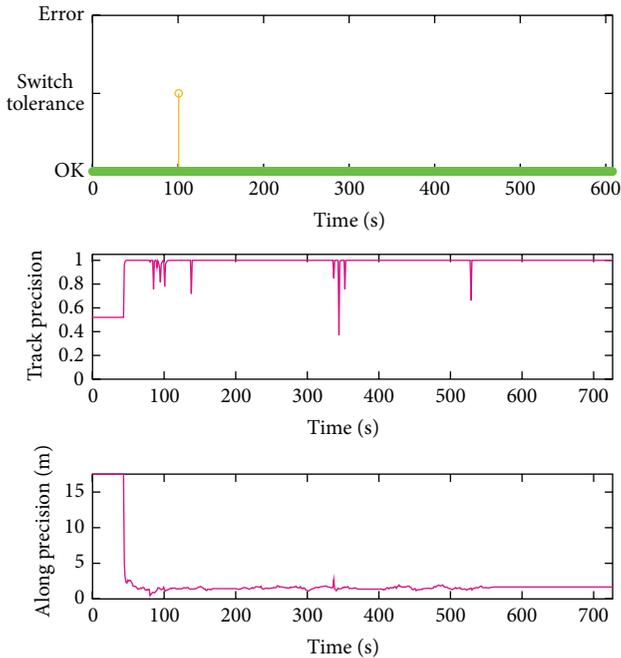


FIGURE 6: Track-selective accuracy and estimation precision of the Bayesian filter approach: RBPF with GNSS position, heading, and IMU yaw rate.

the vicinity of the switch is yellow, and a wrong track is red. There is no track precision shown, as this approach considers no uncertainty but only the nearest track.

Figure 6 shows the accuracy and precision results over time of the realized Bayesian filter with IMU and GNSS of Run 1. At one splitting switch, the filter was estimating an

TABLE 4: Compact track selective results.

Localization method	Track selectivity (parallel tracks) $E_{TS,P}$ (58.5 km)	Switch way res. (107 split switches) $E_{SW}$
(1) Map match, GNSS pos.	78.1%	—
(2) RBPF, GNSS pos.	84.0%	69.2%
(3) (M. 2) + GNSS head	90.2%	96.3%
<b>(4) (M. 2) + IMU yaw rate</b>	<b>99.3%</b>	<b>97.2%</b>
(5) (M. 2) + head + yaw rate	99.2%	96.3%

incorrect track within the tolerance region. The track precision (middle plot) is shortly reduced with the occurrence of split switches as seen in Figure 4. The along precision (bottom plot) is initially coarse but quickly drops after train departure to an average of 1.4 m. This along precision is the weighed empirical deviation of the particle distribution of (37).

**9.2. Cumulative Track Selectivity.** A detailed cumulative evaluation is presented in Table 3. Five localization methods are evaluated, in particular the simple map-matching and four different Rao-Blackwellized particle filter implementations (RBPF). The number of particles is  $N_p = 100$  in all RBPF evaluations. Each accuracy category is shown in percentage of the total distance of 230 km. As all methods solve the single track scenarios, the track accuracy is shown additionally relative to the total distance of multiple track scenarios (58.5 km) in parenthesis. The evaluation results of a standing train periods are disregarded in this table. The error events in Table 3 indicate how often a transition to wrong tracks happened. Depending on the method, this can be traced back to faulty switch resolution. A late switch error represents a resolved switch way after the evaluation threshold, whereas a failed switch relates to a wrong resolved switch way.

The compact results according to (42) and (43) are presented in Table 4. There are no switch way resolution results for Method 1 as the simple map-match considers only the nearest track and does not resolve a switch way.

## 10. Discussion

**10.1. Discussion on Results.** One major goal of train localization is a track-selective estimation result. As seen from

Table 3, Method 1 (reference approach of simple map-matching) has severe problems to determine the correct track at 21.9% of parallel tracks. The proposed algorithm RBPF uses also GNSS positions (Method 2) but shows some improvement with 16.0% of wrong tracks on multiple track scenarios for 33 times. The filtered methods, especially the last three, converge to one track in a parallel track scenario. There, the switch resolution is essential and a failed switch forces the estimate to stay on the wrong track until a merging switch to the true track corrects the estimate again. For example, Method 3 shows an improvement in error events but stays three times on a wrong, parallel track.

The best results are achieved with Method 4 (RBPF with yaw rate) on 99.3% of parallel tracks which relates to an error of  $6.7 \cdot 10^{-3}$  and stands for a false localization on 394 m (24.8 s) in total. With two late and one failed resolved switch way, Method 4 achieves 97.2% in switch way resolution, respective of an error of  $2.8 \cdot 10^{-2}$ .

The definition or adjustment on the switch way evaluation threshold has direct impact on the results and was 50 m. In case this threshold tends to 0 m, the track-selective results over distance can be extracted from the “OK” column of Table 3 and are slightly worse. However, the switch way resolution results of (43) will severely decrease with a zero tolerance area by many late resolved switches.

The results are quite sensitive to the parameters such as measurement noise, the noise ratio of different measurements, resampling occurrence, process, and sampling noise as well as the map quality. Better or even perfect results may be expected by exhaustive optimization of the map and filter parameters. As an unwanted consequence, map and parameters may match this limited data set and the results loose generality. Nevertheless, the following tendencies can be seen from the results: the additional use of the GNSS heading (Method 3) shows only little improvements with 9.8% in terms of errors compared to Method 2 (GNSS position only). Further, the combination of all likelihoods of Method 5 (GNSS position and heading, IMU yaw rate) does not show an improvement. The most likely explanation of this effect might be the map with its coarse heading and curvature geometry. These values are derived from positions and are consequently dependent. Slightly wrong positions cause an error in both values. Because of this dependency the combination of the measurements has no further gain in accuracy.

For the realized implementation of the sequential Bayesian filter, the particle filter approach was chosen, as the particles can sample the different hypotheses and any nonlinear distribution. The Rao-Blackwellization marginalizes the linear state variables and estimates them with a Kalman filter. As a consequence, the particle filter samples only the cross track hypotheses ( $id$ ) and the along positions ( $s$ ). The number of particles are relatively low ( $N_p = 100$ ) as the particles are limited and constraint on the tracks. Additionally, the linear state variables are estimated by nested Kalman filters (Rao-Blackwellization). A particle filter approximates distributions and induces two times additional noise as a tradeoff for convergence reasons: the sampling adds noise, which is

needed to maintain particle diversity as well as the resampling in order to avoid particle depletion and divergence.

*10.2. Comparison.* Lauer and Stein [13] (GNSS, velocity sensor) showed a gain in confidence about the track decision between a simple map-match and the proposed estimation algorithm. A similar gain can be seen between simple map-match (Method 1) and RBPF with GNSS positions only (Method 2). Böhringer [6] received slightly better track-selective results (99.78%) compared to Method 4 (99.33%) with a different data set and an algorithm with additional eddy current sensor as switch way detector. Hensel et al. [12] do not consider explicit figures on switch way resolution. From switch detection and classification rates over 99%, a switch way resolution may be deduced with a similar performance. In comparison, Method 4 reaches 97.2% of correct switch way resolution even without an eddy current sensor and a coarse geometric map from OSM data.

However, a direct comparison between other approaches is not obvious as different data sets and different evaluation metrics are in use. Therefore, comparative results can be seen as quite similar as the differences are marginal for different data sets and metrics. Finally, from the literature and the present results, it can be reasoned that most of the gain in accuracy can be achieved by using an estimation filter as well as using sensors which can measure the competing switch ways. As this is quite expectable, further investigations are necessary to identify the smaller gains and differences of varied filters or sensor fusion approaches on the same data set.

## 11. Proposed Enhancements

Several directions of performance improvements of train localization are identified.

*11.1. Advances of the RBPF.* The particle filter can be improved with advanced procedures for better particle diversity on along track. Secondly, an enhanced resampling timing may be investigated, which suspends resampling near switches for an undisturbed switch way resolution.

*11.2. Improved Odometry.* The along-track odometry can be extended with slope estimation from IMU pitch rate integration, map information about slope, or gravity vector estimation from acceleration measurements. A slope consideration would increase the accuracy of relative along-track estimation and also increase the range to propagate localization in GNSS denied areas.

*11.3. Additional Sensors.* A further increase in track-selective accuracy, outage robustness, and redundancy can be considered with the use of extrinsic sensors, such as magnetic sensors, cameras, LIDAR, or aperture radar with direct switch way measurements.

*11.4. More Accurate Map.* An accurate curvature and heading information in the map is a crucial factor for a correct switch

way resolution with GNSS and IMU. The major advantage of generating a map from an OSM data base is to obtain a track map of a certain railway network size with a sufficient number of tracks. A map generation approach from onboard sensor data is presented in [7, 21].

Furthermore, a direct comparison of alternative methods and sensors may be evaluated with same data sets and evaluation metrics. Studies could investigate the accuracy of different algorithms (e.g., multihypothesis filter versus particle filter), or the accuracy gain of different sensor data integration schemes (e.g., loosely coupled GNSS versus tightly coupled).

## 12. Summary

This paper presents a probabilistic train localization approach with a track-selective evaluation. In contrast to other approaches, this train localization comprises a Rao-Blackwellized particle filter (RBPF), a map of the railway tracks and sensor data of a GNSS receiver (Global Navigation Satellite System), and an IMU (inertial measurement unit). A novel RBPF implementation is presented which estimates the train localization posterior recursively. The Rao-Blackwellization marginalizes the linear state variables and estimates them with a Kalman filter. As a consequence, the particle filter samples only the cross track hypotheses ( $id$ ) and the along positions ( $s$ ). The RBPF estimates directly the topological track coordinates; that is, the particles stay on the tracks. Further, a particle distribution can handle different track hypotheses and other nonlinear distributions. The map contains prior knowledge for the measurement models such as the track geometry data. The RBPF is able to resolve the unknown train-to-track orientation at initialization and can handle forward and backward runs of the train.

A novel evaluation method for track selectivity evaluates the localization results with real data recorded from a regional train. This generic evaluation method can be used to generate more comparable results of different approaches with different sensors and measurement data. Train runs were analyzed over 230 km of tracks with 107 split switches and parallel track scenarios of 58.5 km. Further improvements for a safety-of-life train localization of the special realization are discussed towards higher reliability of track selectivity.

The best combination of RBPF filter with GNSS positions and IMU yaw rates showed a track-selective performance of 99.3% on tracks with multiple tracks in the vicinity and 97.2% of successfully resolved switch ways within the tolerance. The realized RBPF approach with GNSS, IMU, and a track map showed promising results towards a track-selective and continuous train localization even with low-cost sensors and runs in real time.

## Competing Interests

The author declares that there are no competing interests.

## Acknowledgments

The author wants to thank the railway transportation company BRB (“Bayerische Regiobahn”) for the support with the measurements, Andreas Lehner (DLR) for the map, Omar Garcia Crespillo (DLR) for initial software implementations, Stephan Sand (DLR) for proofreading, and Christoph Günther (DLR) for comments.

## References

- [1] T. Strang, M. Meyer zu Hoerste, and X. Gu, “A railway collision avoidance system exploiting ad-hoc inter-vehicle communications and galileo,” in *Proceedings of the 13th World Congress and Exhibition on Intelligent Transportation Systems and Services (ITS '06)*, London, UK, October 2006.
- [2] T. Strang, *Railway Collision Avoidance System*, 2015, <http://www.collision-avoidance.org>.
- [3] O. Heirich, P. Robertson, A. C. García, T. Strang, and A. Lehner, “Probabilistic localization method for trains,” in *Proceedings of the Intelligent Vehicles Symposium (IV '12)*, pp. 482–487, IEEE, Alcalá de Henares, Spain, June 2012.
- [4] O. Heirich, P. Robertson, A. Cardalda Garcia, and T. Strang, “Bayesian train localization method extended by 3D geometric railway track observations from inertial sensors,” in *Proceedings of the 15th International Conference on Information Fusion (FUSION '12)*, pp. 416–423, Singapore, July 2012.
- [5] O. G. Crespillo, O. Heirich, and A. Lehner, “Bayesian GNSS/IMU tight integration for precise railway navigation on track map,” in *2014 IEEE/ION Position, Location and Navigation Symposium, PLANS 2014*, pp. 999–1007, usa, May 2014.
- [6] F. Böhringer, *Gleisselektive ortung von schienenfahrzeugen mit bordautonomer sensorik [Ph.D. dissertation]*, Universität Karlsruhe (TH), Karlsruhe, Germany, 2008.
- [7] C. Hasberg, S. Hensel, and C. Stiller, “Simultaneous localization and mapping for path-constrained motion,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 541–552, 2012.
- [8] O. Plan, *GIS-gestützte verfolgung von lokomotiven im werkbahnverkehr [Ph.D. dissertation]*, University of German Federal Armed Forces, Munich, Germany, 2003.
- [9] K. Gerlach and C. Rahmig, “Multi-hypothesis based map-matching algorithm for precise train positioning,” in *Proceedings of the 12th International Conference on Information Fusion (FUSION '09)*, pp. 1363–1369, Seattle, Wash, USA, July 2009.
- [10] A. Broquetas, A. Comerón, A. Gelonch et al., “Track detection in railway sidings based on MEMS gyroscope sensors,” *Sensors*, vol. 12, no. 12, pp. 16228–16249, 2012.
- [11] O. Heirich, A. Lehner, P. Robertson, and T. Strang, “Measurement and analysis of train motion and railway track characteristics with inertial sensors,” in *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems (ITSC '11)*, pp. 1995–2000, IEEE, Washington, DC, USA, October 2011.
- [12] S. Hensel, C. Hasberg, and C. Stiller, “Probabilistic rail vehicle localization with eddy current sensors in topological maps,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1525–1536, 2011.
- [13] M. Lauer and D. Stein, “A train localization algorithm for train protection systems of the future,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 970–979, 2015.

- [14] J. Wohlfeil, "Vision based rail track and switch recognition for self-localization of trains in a rail network," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '11)*, pp. 1025–1030, IEEE, Baden-Baden, Germany, June 2011.
- [15] R. Ross, "Track and turnout detection in video-signals using probabilistic spline curves," in *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems (ITSC '12)*, pp. 294–299, Anchorage, Alaska, USA, September 2012.
- [16] D. Stein, M. Lauer, and M. Spindler, "An analysis of different sensors for turnout detection for train-borne localization systems," in *Computers in Railways XIV*, vol. 135 of *WIT Transactions on The Built Environment*, WIT Press, 2014.
- [17] C. Rahmig, L. Johannes, and K. Lüddecke, "Detecting track events with a laser scanner for using within a modified multi-hypothesis based mapmatching algorithm for train positioning," in *Proceedings of the European Navigation Conference (ENC '13)*, Vienna, Austria, April 2013.
- [18] C. Fouque and P. Bonnifait, "Matching raw GPS measurements on a navigable map without computing a global position," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 887–898, 2012.
- [19] S. S. Saab, "A map matching approach for train positioning. I. Development and analysis," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 2, pp. 467–475, 2000.
- [20] F. Gustafsson, F. Gunnarsson, N. Bergman et al., "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.
- [21] O. Heirich, P. Robertson, and T. Strang, "RailSLAM—localization of rail vehicles and mapping of geometric railway tracks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '13)*, pp. 5212–5219, IEEE, Karlsruhe, Germany, May 2013.
- [22] Open Street Map (OSM), 2015, <http://www.openstreetmap.org>.
- [23] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, Ganga-Jamuna Press, Lincoln, Mass, USA, 2006.
- [24] A. Doucet, N. De Freitas, K. Murphy, and S. Russell, "Raoblackwellised particle filtering for dynamic bayesian networks," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, Stanford, Calif, USA, June-July 2000.
- [25] F. Gustafsson, *Statistical Sensor Fusion*, Studentlitteratur, Lund, Sweden, 2012.
- [26] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [27] F. Daum and J. Huang, "Curse of dimensionality and particle filters," in *Proceedings of the IEEE Aerospace Conference*, pp. 1979–1993, March 2003.

## Research Article

# A Novel Improved Probability-Guided RANSAC Algorithm for Robot 3D Map Building

Songmin Jia,<sup>1,2,3</sup> Ke Wang,<sup>1,2,3</sup> Xiuzhi Li,<sup>1,2,3</sup> and Tao Xu<sup>1,2,3,4</sup>

<sup>1</sup>College of Electronic and Control Engineering, Beijing University of Technology, Beijing 100124, China

<sup>2</sup>Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China

<sup>3</sup>Engineering Research Center of Digital Community, Ministry of Education, Beijing 100124, China

<sup>4</sup>School of Mechanical and Electrical Engineering, Henan Institute of Science and Technology, Xinxiang 453003, China

Correspondence should be addressed to Songmin Jia; [jsm@bjut.edu.cn](mailto:jsm@bjut.edu.cn)

Received 4 August 2015; Revised 19 February 2016; Accepted 16 March 2016

Academic Editor: Maan E. El Najjar

Copyright © 2016 Songmin Jia 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 a novel improved RANSAC algorithm based on probability and DS evidence theory to deal with the robust pose estimation in robot 3D map building. In this proposed RANSAC algorithm, a parameter model is estimated by using a random sampling test set. Based on this estimated model, all points are tested to evaluate the fitness of current parameter model and their probabilities are updated by using a total probability formula during the iterations. The maximum size of inlier set containing the test point is taken into account to get a more reliable evaluation for test points by using DS evidence theory. Furthermore, the theories of forgetting are utilized to filter out the unstable inliers and improve the stability of the proposed algorithm. In order to boost a high performance, an inverse mapping sampling strategy is adopted based on the updated probabilities of points. Both the simulations and real experimental results demonstrate the feasibility and effectiveness of the proposed algorithm.

## 1. Introduction

RANSAC algorithm is one of the popular methods for sensor data registration and modeling. In some vision-based SLAM (Simultaneous Localization and Mapping) algorithms, RANSAC algorithm provides an efficient solution for image matching procedure and establishes the data association among different views [1–3]. There are two typical types of algorithm for image matching, the dense way [4, 5] and sparse way [6, 7]. In dense match approach, the whole image is used for parameter estimation [8, 9]. Although this method is quite robust, it may be inaccurate when occlusion regions exist in the matching images. Sometimes, the influence of occlusions is reduced by using a robust weighted cost function. In sparse match step, image features are detected by using SIFT, SURF, or any other feature detection algorithms. The features are matched by using the distance of feature descriptor, and the matching pairs are sometimes ambiguous. To efficiently achieve a correct matching result, some robust algorithms

were adopted to remove mismatching pairs, such as M-estimation [10], LMedS (Least Median of Squares) [11], or RANSAC (Random Sample Consensus) [12] algorithm. M-estimation established a new cost function with a robust weight. It worked well in some cases but was vulnerable to the noise. LMedS optimized the model by minimizing the median of errors. When the outlier rate was larger than 50%, M-estimation and LMedS might be no longer applicable. With the advantages of easy implementation and strong robustness, RANSAC algorithm was widely used in model parameter estimation problem. In standard RANSAC algorithm, a hypothesis set was randomly selected to estimate a parameter model. And an inlier set was detected by testing all input data with the estimated parameter model. A maximum size of inlier set was expected to be found within a predetermined iteration. However, the performance of this standard RANSAC was sometimes low. Even worse was the fact that the solution may not be reached when all iterations were finished.

To efficiently improve the performance of standard RANSAC algorithm, some methods have been proposed in recent decades. A hypothesis evaluation function and local optimization procedure were adopted to achieve a more accurate result. MSAC (M-estimation SAC) [10] evaluated the test point set with a bounded loss function to achieve a maximum likelihood consensus set. MLESAC (Maximum Likelihood SAC) [10] evaluated the hypothesis set by using the probability distribution of errors. The inlier error was modeled with an unbiased Gaussian distribution and outlier error used a uniform distribution. The maximum likelihood estimation was solved by minimizing a cost function. MAPSAC (Maximum A Posteriori Estimation SAC) [13] followed a Bayesian approach to solve the RANSAC problem with a robust MAP estimation. LO-RANSAC (Local Optimized RANSAC) [14] adopted an inner model reestimation procedure to improve the accuracy of the RANSAC algorithm.

Moreover, some heuristic mechanism sampling strategies and partial evaluation procedures were adopted to speed up the convergence of the algorithm. It seemed that a good sample strategy will reduce the time cost which was spent in finding the solution. A hypothesis set was selected based on the probabilities of test points in the Guided MLESAC [15]. In PROSAC (Progressive SAC) [16], the matching score was used as a prior knowledge for sorting the test data. A hypothesis set was selected among the data which was in the top-ranked matching score. It was also progressively tested on the less ranked data. In some extreme cases, the whole data would be tried in this algorithm. According to the assumption that an inlier tends to be closer to the inliers, NAPSAC (N Adjacent Points SAC) [17] sampled the data within a defined radius around a selected point. Based on the preliminary test of the hypothesis, Chum and Matas proposed R-RANSAC (Randomized RANSAC) [18] and R-RANSAC SPPR (Sequential Probability Ratio Test) [19]. These two methods performed a preliminary test based on  $T_{d,d}$  and SPPR test after evaluating test points in every iteration. And a full test procedure was performed only when a hypothesis set passed the preliminary test. The preliminary test procedure effectively removed the obvious mistakes of hypothesis sets and improved the efficiency of RANSAC algorithm. Optimal RANSAC [20] adopted an inlier sample procedure to achieve a more accurate model estimation. When the size of the current inlier set was larger than a threshold, an inlier sample procedure would be performed to achieve a more reliable solution in the inlier set.

Furthermore, some intelligent algorithms such as Genetic Algorithm (GA) and multilayered feed-forward neural networks (NFNN) were also proposed in RANSAC algorithm. Rodehorst proposed a novel RANSAC algorithm based on GA [21]. In GASAC, the parents were generated by a standard RANSAC algorithm with a robust cost evaluation. Then, the best solution was achieved by using crossover and mutation operators on parents in GA procedure. Moumen presented a rather comprehensive study of robust supervised training of MFNN in a RANSAC framework from the standpoint of both accuracy and time [22]. In the iteration of RANSAC, the parameter model was estimated by using a small MFNN which was minimizing the mean squared error (MSE) with

a standard back propagation algorithm. All inlier points were used to reestimate a new parameter model by training a new MFNN. And a new hypothesis set was achieved by using this new MFNN model. The convergence solution was achieved until the inlier set did not change any more.

In this paper, we propose a novel improved probability-guided RANSAC (IPGSAC) algorithm for mobile robot 3D map building. Under the framework of standard RANSAC algorithm, two types of probabilities are evaluated for test points by using a total probability formula and the statistics of maximum size of inlier set. To achieve a more robust evaluation of test points, DS evidence theory [23] is adopted to synthesize the multisource evaluation of test points. Moreover, the theories of forgetting are employed to filter out the unstable inliers. Based on the probability of test points, an inverse mapping sampling strategy is utilized to improve the convergence rate of the proposed algorithm. Finally, this proposed IPGSAC algorithm is applied for the mobile robot 3D map building. All the experimental results show the feasibility and effectiveness of the proposed algorithm.

The rest of this paper is organized as follows: in Section 2, we summarize IPGSAC before explaining each in detail. The components of our robot map building procedure are detailed in Section 3. Our simulation and real experimental results are described in Section 4. Finally, we give our conclusions and future work in Section 5.

## 2. IPGSAC Algorithm

*2.1. Methodology Overview.* The proposed IPGSAC algorithm is illustrated in Figure 1. At the beginning of IPGSAC algorithm, the probabilities of test points are initialized with a hybrid distribution. Based on those probabilities,  $n$  points are selected for model estimation by employing an inverse mapping sampling strategy. The inlier and outlier sets are distinguished with a tolerance threshold  $r_0$ . When the residual error  $r_i$  of point  $p_i$  is larger than  $r_0$ , the  $p_i$  is identified as an outlier point. Then, two types of probabilities are evaluated for test points by using a total probability formula and the statistics of maximum size of inlier set. According to the average observations of inlier points, the theories of forgetting are employed to reduce the redundancy of unstable inliers. To achieve a more reliable evaluation, the probability evaluations are synthesized by using DS evidence theory. When the maximum iteration limit is arrived at or the stopping criterion is reached, the main loop of IPGSAC will be finished. Finally, we reestimate the model parameter by using all inliers to achieve a more reliable inlier set with 3~5 iterations. In the standard RANSAC, the minimum number of iterations  $k_{\max}$  ensures that a correct hypothesis set is achieved with a determined confidence level at least once and it can be estimated by

$$k_{\max} = \frac{\log(1-z)}{\log(1-w^n)}, \quad (1)$$

where  $z$  is the confidence level,  $w$  indicates the inlier rate, and  $n$  indicates the minimum number of test points for model estimation. In the proposed IPGSAC algorithm,

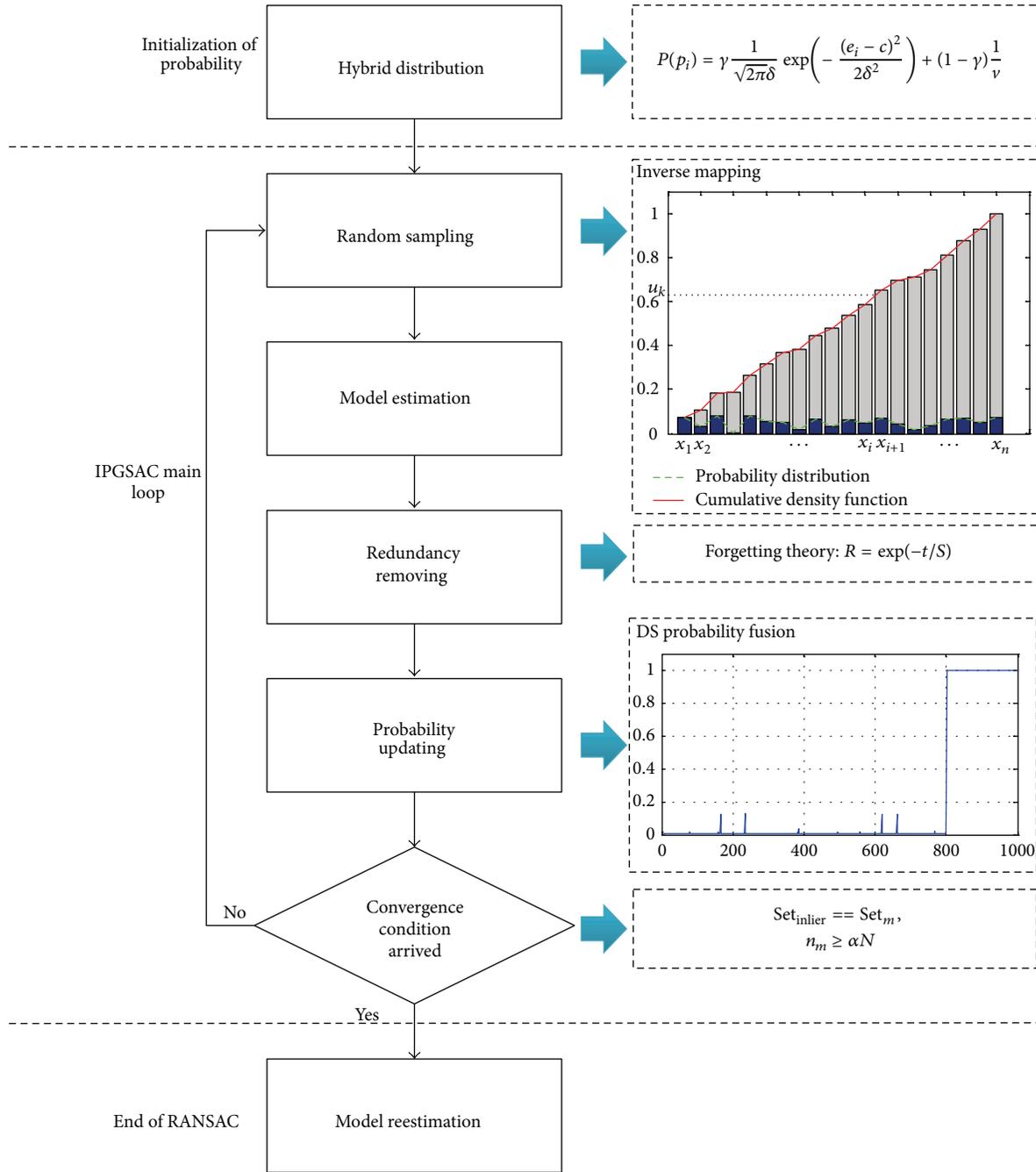


FIGURE 1: The illustration of the proposed IPGSAC algorithm.

the maximum number of iterations is limited by (1) with the confidence level 98%.

In this proposed algorithm, a maximum inlier set is expected to be found when the algorithm is converged. To avoid falling into local optimum, a pseudo inlier rate is used to limit the minimum size of inlier set. The stopping criterion is described as follows:

$$\begin{aligned} \text{Set}_{\text{inlier}} &== \text{Set}_m, \\ n_{\text{inlier}} &\geq \alpha N, \end{aligned} \tag{2}$$

where  $\text{Set}_{\text{inlier}}$  is the inlier set which is acquired in the current iteration,  $n_{\text{inlier}}$  is the number of points in  $\text{Set}_{\text{inlier}}$ ,  $\text{Set}_m$  indicates the maximum inlier set so far, and  $\alpha$  is the pseudo minimum inlier rate.  $N$  is the total number of test points. Operator “==” means that two sets are very similar. The detailed description of each component of the proposed algorithm is given in the following sections.

**2.2. Initialization of Probabilities.** Inspired by MLESAC algorithm, we assume that the inlier data is satisfied to a normal

distribution and the outlier data is fitted to a uniform distribution. The hybrid distribution is described as follows:

$$P(p_i) = \gamma \frac{1}{\sqrt{2\pi}\delta} \exp\left(-\frac{(e_i - c)^2}{2\delta^2}\right) + (1 - \gamma) \frac{1}{\nu}, \quad (3)$$

where  $e_i$  is the residual error of  $p_i$  with the model estimated by using all data,  $c$  is the expectation of inlier errors,  $\delta$  is the variance of  $e_i$ ,  $\nu$  determines the probability of outlier point, and  $\gamma$  indicates whether  $p_i$  is an inlier or not. When  $p_i$  is an inlier point, then  $\gamma = 1$ ; otherwise  $\gamma = 0$ . To evaluate the distribution, the length of error interval  $L$  is calculated with the minimum and maximum of error. The mean and variance of error are used as an initial estimation of  $c$  and  $\delta$ , respectively. By limiting the length of searching interval with the maximum value of  $gL$  and  $2\delta$ ,  $c$  and  $\delta$  are solved by using a standard normal distribution with several iterations. In this proposed algorithm,  $\nu = N$  and  $g = 0.1$ .

**2.3. Evaluation Inlier Probabilities.** In standard RANSAC algorithm, the parameter model is estimated by using a random sample test set. According to this estimated model, the inlier and outlier set are determined in this iteration. While this procedure should be aimless, it needs a large number of iterations to find an optimal solution. This paper aims to provide an effective and reliable inlier evaluation method and speed up the convergence of RANSAC algorithm. In the BaySAC [24], Botterill et al. assumed that the probabilities of test points were independence in the same hypothesis set. Firstly,  $n$  points were selected to estimate a parameter model  $s$ . Then, all input points were tested to find an inlier set. The probabilities of inlier points were updated by using (4) and did not change the outlier's probabilities:

$$\begin{aligned} P(p_i \in M | s \notin M) &= \frac{P(p_i \in M) P(s \notin M | p_i \in M)}{P(s \notin M)} \\ &= \frac{P(p_i \in M) (1 - P(s \subseteq M | p_i \in M))}{1 - P(s \subseteq M)} \\ &= \frac{P(p_i \in M) (1 - P(s \subseteq M) / P(p_i \in M))}{1 - P(s \subseteq M)} \\ &= \frac{P(p_i \in M) - P(s \subseteq M)}{1 - P(s \subseteq M)}, \end{aligned} \quad (4)$$

where  $M$  is the consensus set that contains all inlier points and all correct models and  $p_i$  indicates a test point.  $P(p_i \in M)$  evaluates the probability of a point being an inlier.  $P(p_i \in M | s \notin M)$  describes the probability of a point being an inlier, under the condition that the estimated model  $s$  is not a subset of  $M$ . The BaySAC is poor when the probabilities of test points are the same. With the proposed initialization of probability, it is sometimes difficult to achieve the convergence of BaySAC. In order to improve the convergence of the proposed algorithm, a multisource evaluation strategy is employed to achieve a higher confident probability analysis of test points.

First of all, a full probability update procedure is used to achieve a more reliable and comprehensive evaluation of test points. The total probability update procedure is expressed as

$$\begin{aligned} P(p_i \in M) &= P(p_i \in M | s \notin M) P(s \notin M) \\ &\quad + P(p_i \in M | s \subseteq M) P(s \subseteq M), \end{aligned} \quad (5)$$

where  $P(p_i \in M | s \notin M)$  is calculated by using (4) and  $P(p_i \in M | s \subseteq M)$  indicates the probability of  $p_i$  being an inlier, under the condition that the estimated model  $s$  is a subset of  $M$ :

$$P(p_i \in M | s \subseteq M) = 1 - \left| \frac{r_i}{r_0} \right|, \quad (6)$$

where  $r_i$  is the residual error of point  $p_i$  with the current estimated model.  $P(s \subseteq M)$  evaluates the probability of the estimated model being a correct model. As the number of the inliers illustrates the correctness of the estimated model,  $P(s \subseteq M)$  is estimated by using the proportion of inlier in this iteration:

$$\begin{aligned} P(s \subseteq M) &= \frac{n_{\text{inlier}}}{N}, \\ P(s \notin M) &= \frac{n_{\text{outlier}}}{N}, \end{aligned} \quad (7)$$

where  $n_{\text{inlier}}$  and  $n_{\text{outlier}}$  are the number of inlier and outlier points, respectively.  $N = n_{\text{inlier}} + n_{\text{outlier}}$  is the total number of test points. Moreover,  $n_{\text{inlier}}$  not only indicates the correctness of current model estimation, but also evaluates the possibility that the test point is an inlier. The statistics of maximum size of inlier set which contains the test points is employed to reevaluate the probabilities of test points in the iteration. The evaluation of inlier and outlier points are defined as

$$\begin{aligned} P_m(p_i \in M) &= \frac{n_i}{\tilde{n}}, \\ P_m(p_i \notin M) &= 1 - \frac{n_i}{\tilde{n}}, \end{aligned} \quad (8)$$

where  $n_i$  is the maximum size of inlier set which contains the test point  $p_i$  and  $\tilde{n} = \sum_{i=1}^N P(p_i \in M)$  indicates the current predicted size of inlier set.

**2.4. Probability Fusion with DS Evidence Theory.** During the iteration of IPGSAC algorithm, the probabilities of test points are updated by using our evaluation criteria. As it is a type of uncertain information, the probabilities indicate the possibility that a point belongs to an inlier set. Generally, the information is characterized as inherently uncertain, imprecise, and incomplete. The representation and reasoning of uncertainty information are one of the important research issues in artificial intelligence (AI). Many researchers devote themselves to modeling the representation and reasoning of the uncertain information. According to the inherently uncertain characteristics of information, a more reasonable and accurate evaluation is achieved by synthesizing some different sources of information with some specified rules.

DS evidence theory, proposed by Dempster and Shafer, is one of the popular methods for the multisource information fusion. The evidence theory describes the uncertainty of information by using the discernment framework, the basic probability belief functions, and likelihood functions [23]. Suppose that there is a problem written as  $x$  and the discernment set  $\Theta$  contains all the possible solutions of  $x$ . All the elements are mutually independent in the discernment set  $\Theta$ . The solutions of  $x$  only take from  $\Theta$  in any case.  $\Theta$  is defined as a discernment framework and written as  $\{\theta_1, \theta_2, \dots, \theta_i, \dots, \theta_n\}$ .  $\theta_i$  is a really defined focal element of discernment framework  $\Theta$ ;  $n$  is the number of elements in  $\Theta$ . The power set of a discernment framework  $2^\Theta$  contains all subsets of the discernment framework. There are  $2^n$  elements in  $2^\Theta$  and each of the elements corresponds to a solution for  $x$ . The power set  $2^\Theta$  is composed of the following subsets:

- (1) It contains the empty set  $\phi$  and all of the elements in  $\Theta$ ;
- (2) If subset  $A, B \in 2^\Theta$ , then  $A \cup B \subseteq 2^\Theta$ ;
- (3) Excepting (1) and (2), it no longer contains any other subsets.

In DS evidence theory, a probability belief function is adopted to describe the uncertainty of evidence. This probability belief function is defined as  $m : 2^\Theta \rightarrow [0, 1]$ , such that

$$\begin{aligned} \sum_{A \in \Theta} m(A) &= 1, \\ m(\phi) &= 0, \end{aligned} \quad (9)$$

where  $\phi$  is an empty set,  $m$  is a probability belief function in  $2^\Theta$ , and  $m(A)$  describes the confidence probability of solution  $A$ .

Generally, the probability belief function is used to evaluate the probability of a reasoning that endows the supposing of the solutions. The final judgment is generally influenced by many factors. According to the DS evidence theory, a synthesis method is illustrated to fuse two or more basic probability belief functions by using a sum of orthogonal operations. Assume that  $m_1$  and  $m_2$  are the corresponding probability belief functions which are mutually independent in the same discernment set  $\Theta$ . For the focal elements  $A_1, A_2$  and  $B_1, B_2$ , the synthesis operation is described as

$$\begin{aligned} K &= \sum_{A_i \cap B_j = \phi} m_1(A_i) m_2(B_j) \\ m(A) &= \begin{cases} \frac{\sum_{A_i \cap B_j = A} m_1(A_i) m_2(B_j)}{1 - K} & \forall A \subset \Theta, A \neq \phi \\ 0 & A = \phi, \end{cases} \end{aligned} \quad (10)$$

where  $K$  is conflict coefficient and  $1/(1 - K)$  is regularization factor.

In this proposed IPGSAC algorithm, the status of test samples is assigned as inlier( $A$ ) or outlier( $B$ ) and the discernment framework  $\Theta$  is  $\{A, B\}$ . The probability brief function

is written as  $[m(A_i), m(B_i)]$  for each test point. According to Section 2.3, the probability brief function is defined as

$$\begin{aligned} m_1(A) &= P(p_i \in M), \\ m_1(B) &= P(p_i \notin M), \\ m_2(A) &= P_m(p_i \in M), \\ m_2(B) &= P_m(p_i \notin M). \end{aligned} \quad (11)$$

**2.5. Probability Filtering with Theories of Forgetting.** For some low constrained model estimation problem, there exist a few of outlier points which easily satisfy the current estimated model but are not stable. During the DS fusion procedure, this incorrect judgment will have a great influence on the convergence of the proposed algorithm. Even worse is the fact that the sample probability of inlier is finally much lower than the outlier and the algorithm is hardly converged. To further improve the proposed algorithm, the theories of forgetting are used to filter out the unstable inliers and achieve a more stable solution.

According to theories of forgetting, the forgetting curve can be roughly described as

$$R = \exp\left(-\frac{t}{S}\right), \quad (12)$$

where  $R$  is memory retention,  $S$  is the relative strength of memory, and  $t$  describes the time. As our proposed algorithm is an online learning procedure, we use (13) to describe the forgetting and reviewing procedure:

$$R = \exp(o - 1), \quad (13)$$

where  $o$  indicates the average observations when the inlier set is similar to current maximum set. And the maximum inlier size of  $p_i$  is refreshed by  $n_i = Rn_i$ .

Figure 2 shows the estimated parameters for the fundamental matrix estimation when the outlier rate is 80%. Figure 2(a) demonstrates a typical online learning procedure with 50 iterations. During the iteration, the probabilities of test points are updated by using a total probability formula. After 30 iterations, a convergent solution is achieved with a stable sample probability of inlier and outlier. In Figure 2(b), the maximum inlier size of point is adopted to reinforce the probabilities of test points by using proposed DS fusion procedure. Due to the incorrect judgment of test points, the probability of outlier is waved between 0 and 0.8. And the sample probability of inlier is decreased after several iterations. Although a correct inlier set is achieved in the 3rd iteration, it becomes more and more difficult to reach the convergence of the algorithm. Figure 2(c) illustrates the result of our proposed algorithm which is guided by proposed probability update procedure. After a correct hypothesis set is found, the probabilities of test points obviously distinguished that the probability of inlier is near to 1 and outlier is near to 0. The influence of the incorrect judgment is filtered by using the theories of forgetting. After a few iterations, the curve of the predicted inlier size is much smoother than illustrated in

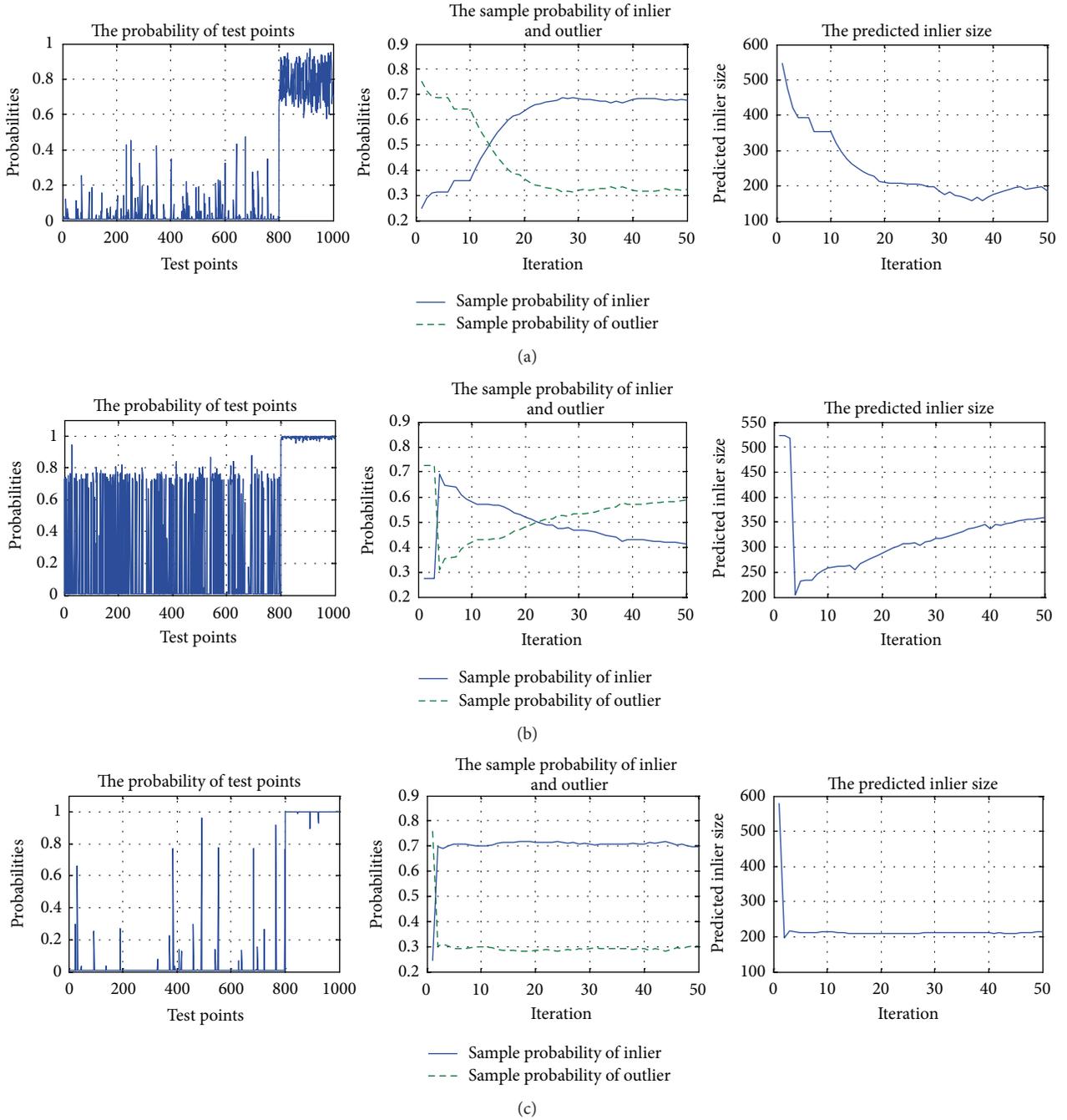


FIGURE 2: The estimated parameters during the fundamental estimation simulation. (a) shows the results by using the total probability formula, (b) illustrates the result with DS evidence theory, and (c) demonstrates the result of the procedure of the proposed algorithm.

Figures 2(a) and 2(b) and it is highly consistent with the presetting of the simulation. The results demonstrate the high performance and reliability of the proposed algorithm.

**2.6. Inverse Mapping Sampling Strategy.** During the procedure of RANSAC, a good sampling strategy is an important guarantee of convergence of algorithm. In IPGSAC, the probabilities of test points are evaluated and merged by using DS evidence theory. To speed up the convergence rate of the proposed algorithm, an inverse mapping sampling strategy

[25, 26] is utilized for the hypothesis set selection according to the updated probability of test points. The principle of this sampling procedure is as follows: assume that the probability distribution function of a continuous random variable  $X$  is written as  $F(x)$ .  $F^{-1}(u)$  is an inverse function of  $F(x)$  where  $u \in [0, 1]$ .  $F^{-1}(u)$  and  $F(x)$  have the same probability distribution function and the probability distribution is expressed as

$$P(X \leq x) = P(F^{-1}(u) \leq x) = P(U \leq F(x)). \quad (14)$$

For a discrete random variable  $X$ , a cumulative probability distribution function is expressed as

$$P(X \leq x) = \sum_{x_i \leq x} P(x_i). \quad (15)$$

In the inverse mapping sampling strategy,  $[0, 1]$  is divided into  $n$  subsections which is corresponding to sample points. The  $k$ th subsection is expressed as  $[\sum_{i=1}^{k-1} P(x_i), \sum_{i=1}^k P(x_i)]$ . When a uniform sampled variable  $u_k \in [0, 1]$  is achieved in the  $i$ th subsection, the corresponding data of this subsection is the expected output. In our proposed algorithm, the probability of test point  $p_i$  is normalized by  $w_i = P(p_i) / \sum_{i=1}^n P(p_i)$ . And a sampling cumulative density function (CDF)  $F$  is defined by  $F(p_i) = \sum_{k=1}^i w_k$ . The expected output is  $p_i$  when the random variable  $u_k$  is in the subsection  $[F(p_i), F(p_{i+1}))]$ .

In our proposed IPGSAC algorithm, a hypothesis set is randomly selected by using the inverse mapping sampling strategy with the probabilities of test points. When the probabilities of test points are equal, the sampling method obeys a uniform sampling approach and it is consistent with the standard RANSAC algorithm. When those probabilities are different, the procedure will improve the sample probability of the point which has a high probability. When a correct hypothesis set is achieved, especially, the probabilities of inlier will be greatly improved with the proposed evaluation procedure. Then, the sampling probability of these inliers will be increased rapidly. Therefore, it will effectively speed up the convergence rate of the proposed algorithm.

**2.7. Implementation of IPGSAC Algorithm.** A general framework of the proposed IPGSAC algorithm is shown in Algorithm 1. The proposed algorithm can be divided into the initialization of probability, the main loop, and the final reestimation. At the beginning of this algorithm, some parameters should be defined before the procedure of each part. In the initialization step, the test points' probabilities are evaluated by the hybrid distribution. Then, the main loop of the proposed algorithm starts to find the solution of model estimation. In this procedure, a hypothesis set  $\text{Set}_{\text{test}}$  containing  $n$  test points is randomly selected for model estimation by using the inverse mapping sampling strategy. When the model estimation error is larger than the tolerance threshold  $r_0$ , the model estimation fails and the probabilities of test points are updated by using (4) with  $P(s < M) = 0$ . When the model estimation succeeds, the probabilities of test points are evaluated by using the total probability formula. And the maximum inlier size of test point in  $\text{Set}_{\text{inlier}}$  is refreshed by current inlier size  $n_{\text{inlier}}$ . To reduce the influence of incorrect judgment, a forgetting factor  $R$  is computed with the average of observation when an inlier set is similar to current maximum set. Other evaluations of test points are estimated with the maximum inlier size  $n_i$  and forgetting factor. A more reliable probability evaluation of test points is achieved by employing the DS evidence theory. Finally, the convergent constraint should be checked in every iteration. When the convergent constraint arrives, the reestimation step is performed in several times to achieve a more reliable inlier set and model estimation.

### 3. Robot 3D Map Building Using IGPSAC

**3.1. Preliminaries Definition.** Kinect is a low cost depth sensor which is widely used in 3D map building system. Assume that the image and depth captured by Kinect are precalibrated by using the method presented in [27] and the depth is pre-warped into the image frame. The intrinsic parameter matrix  $K$  of RGB camera is written as

$$K = \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}, \quad (16)$$

where  $(f_u, f_v)$  and  $(c_u, c_v)$  are the focal and image center of RGB cameras, respectively. According to these intrinsic parameters, the projection from a 3D point to image pixel is described as

$$u = K\pi(p), \quad (17)$$

where  $p = (x, y, z)$  is a 3D point in current camera frame and  $\pi(p) = (x/z, y/z)$ .

During the map building procedure, Kinect is fixed on a mobile robot with a consistent translational speed. We assume that the pose of robot is the same as the Kinect and the world coordinate is set at the location where robot starts to run. At the time  $i$ , robot's pose is described as

$$T_i = \begin{bmatrix} R_i & t_i \\ 0 & 1 \end{bmatrix}, \quad (18)$$

where  $T_i \in \text{SE}(3)$ ;  $\text{SE}(3)$  is a special rigid transformation group.  $R_i$  is the rotation matrix;  $t_i$  is the translation vector. According to the *Lie* algebra,  $T_i$  is expressed as

$$T_i = \exp(\tilde{\zeta}), \quad (19)$$

where  $\zeta = (\alpha, \beta, \gamma, t_x, t_y, t_z)$ , and  $\tilde{\zeta}$  is define as

$$\tilde{\zeta} = \begin{bmatrix} 0 & \gamma & -\beta & t_x \\ -\gamma & 0 & \alpha & t_y \\ \beta & -\alpha & 0 & t_z \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (20)$$

**3.2. Pose Estimation Based on IPGSAC.** In this paper, FAST corner algorithm [28, 29] is employed to detect the features and the BRIEF [30] descriptor is utilized for feature matching. With the image matching pairs, a derivative ICP [31] algorithm is adopted based on the projection errors. Generally, there exist some invalid depth measurement regions in the captured depth images. Suppose that all matching pairs mentioned in this section have the valid depth measurement. The pose estimation model is described as

$$E = \arg \min_{T_k} \frac{1}{N} \sum_{\Omega} \|r_i(\xi)\|, \quad (21)$$

```

(1) Parameter setting: input data  $Data$ ,  $\alpha = 0.05$ ,
    minimum model estimation data size  $n$ , tolerance threshold  $r_0$ ,
    setup Model function  $modelFunc$ ,  $modelInlier$ ,
    reset maximum inlier size  $p_i$ ,  $n_i = n$ ;
(2) Estimate the maximum iterations  $k_{max}$ ;
(3) Initialize the probabilities of test points by using a hybrid distribution;
(4) for  $i = 1$  to  $k_m$  do
(5)    $Set_{test} = inverse\_sampling(Data, n)$ ;
(6)    $(error, s) = modelFunc(Set_{test})$ ;
(7)   if  $error > r_0$ 
(8)     Update probabilities of points in  $Set_{test}$ ,  $P(p_i \in M) = P(p_i \in M | s \notin M)$ ;
(9)   else
(10)     $(Set_{inlier}, r, n_{inlier}) = modelInlier(Data, s)$ ;
(11)    Update the probabilities of all test points  $P(p_i \in M)$  by using (5);
(12)    Update the maximum inlier size. For  $p_i$  in  $Set_{inlier}$ ,  $n_i = \max(n_i, n_{inlier})$ ;
(13)  end if
(14)  Compute the forgetting factor  $R$ ;
(15)  Compute  $P_m(p_i \in M) = R(p_i)n_i/\tilde{n}$ ;
(16)  Update the probability  $P(p_i \in M)$  using DS evidence theory;
       $K = P(p_i \in M)P_m(p_i \notin M) + P(p_i \notin M)P_m(p_i \in M)$ 
      
$$P(p_i \in M) = \frac{P(p_i \in M)P_m(p_i \in M)}{1 - K}$$

(17)  if  $n_{inlier} > \alpha N$  and  $Set_m == Set_{inlier}$ 
(18)    break;
(19)  end if
(20)  if  $n_{inlier} > maxInlierSize$ 
(21)    Update  $Set_m$  with  $Set_{inlier}$ ;
(22)     $maxInlierSize = n_{inlier}$ ;
(23)  end if
(24) end for
(25) for  $i = 1$  to  $k$  do
(26)   $(error, s) = modelFunc(Set_m)$ ;
(27)   $(Set_m, r, n_i) = modelInlier(Data, s)$ ;
(28) end for

```

ALGORITHM 1: IPG SAC algorithm.

where  $r_i$  is the projection error and defined as  $r_i(\xi) = K\pi(T(\xi)p_i) - x_i$ ;  $p_i$  is a point in the reference point cloud which is corresponding to the new captured feature  $x_i$ .  $\Omega$  indicates a set of image matching pairs;  $N$  is the size of  $\Omega$ .

To achieve the optimization of (21), the projection error  $r_i$  is approximated by first-order Taylor expansion

$$r_i = r_i(0) + J\xi, \quad (22)$$

where  $J = (\partial r_i / \partial \xi)|_{\xi=0}$ .

According to Euler-Lagrange formula, the optimal solution of (21) is solved by using a Newton iteration step with

$$\begin{aligned} \frac{\partial E}{\partial \xi} &= \frac{1}{N} \sum_{\Omega} J^T (r_i(0) + J\xi), \\ \xi &= -\sum_{\Omega} (J^T J)^{-1} \sum_{\Omega} J r_i(0). \end{aligned} \quad (23)$$

Due to the ambiguity of image matching procedure, the proposed IPGSAC algorithm is utilized for removing the outliers (mismatching pairs) and achieving an accurate pose estimation. The detail of this pose estimation procedure is similar to Algorithm 1.

### 3.3. Keyframe Selection with Maximum Matching Distance.

With the accurate estimated poses, a 3D Map is built with the captured depth and color data. Considering that there are about 300,000 points in every frame, it is necessary to create a more concise representation of the 3D map. A keyframe selection mechanism is employed with maximum matching distance in the algorithm. The map building procedure will start only when a new keyframe is captured. However, the accuracy of pose estimation is largely dependent on the overlap of the input data. Therefore, a distance threshold  $d_0$  and an angle threshold  $\theta_0$  are employed for the keyframes selection. In the procedure of keyframe selection, if the pose estimation between current frame  $I_{i+1}$  and keyframe  $I_{kf}^{k-1}$  fails or the distance  $d$  or the angle  $\theta$  between them is larger than the thresholds, the frame  $I_i$  will be chosen as a new keyframe. The principle of keyframe selection is described as follows:

$$I_{kf}^k = \{I_i \mid icp(I_{kf}^{k-1}, I_i) > 0, icp(I_{kf}^{k-1}, I_{i+1}) \leq 0\}, \quad (24)$$

where  $icp(I_1, I_2)$  evaluates the procedure of pose estimation and  $icp = 0$  when pose estimation fails or  $d \geq d_0$  or  $\theta \geq \theta_0$ ; for others,  $icp = 1$ .

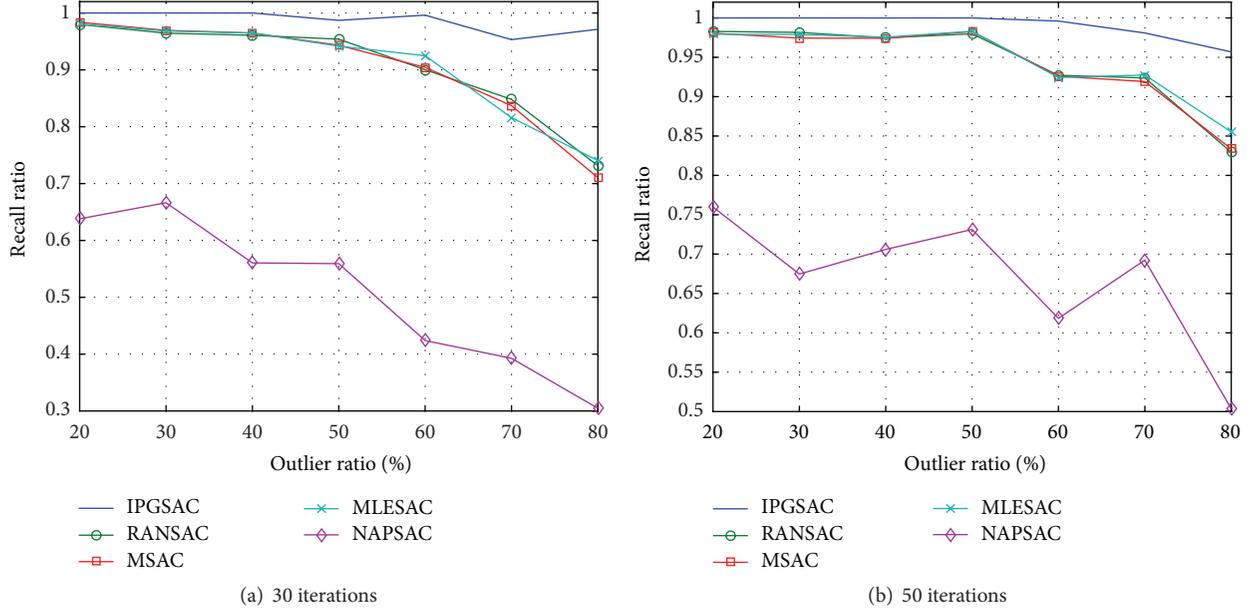


FIGURE 3: The recall ratios with 30 and 50 iterations in line fitting simulation.

**3.4. Loop Detection and Optimization.** During the map building, the quality of global map is largely dependent on the accuracy of the pose estimation. However, a drift arises with the accumulation of the pose estimation error during the 3D map building experiment. To effectively achieve the global optimization of the 3D map, a vision-based loop detecting method is used to check the loop closure when a new keyframe is captured. During the loop detecting procedure, the relative poses between current keyframe and previous nonadjacent keyframes are estimated by using the pose estimation algorithm. A successful estimation of relative pose means that a loop closure is detected and the optimization algorithm will be performed immediately. In this paper, TORO (tree-based network optimizer) [32, 33] algorithm is employed to achieve a global optimization of the generated map. In TORO algorithm, a tree structure is constructed by using the robot's poses and relative poses among keyframes. The error function which evaluates the observed and calculated relative pose is minimized to optimize all the estimated poses and achieve a global consistent 3D map. The optimization function is described as

$$T_{k_f} = \arg \min_{T_{k_f}} F(T_{k_f}),$$

$$F(T_{k_f}) = \sum_{(i,j) \in G} e(T_{k_f}^i, T_{k_f}^j, T_{ij})^T \Lambda e(T_{k_f}^i, T_{k_f}^j, T_{ij}), \quad (25)$$

where  $T_{k_f} = \{T_{k_f}^1, \dots, T_{k_f}^i, \dots, T_{k_f}^n\}$  is the vector descriptor of robot's poses,  $T_{ij}$  is an observed relative pose between  $i$ th keyframe and  $j$ th keyframe,  $\Lambda$  represents, respectively, the mean and the information matrix of a constraint relating the poses, and  $G$  is a relative tree.  $e(T_{k_f}^i, T_{k_f}^j, T_{ij})$  is an error function that measures how well the poses satisfy the observed constraint  $T_{ij}$ .

## 4. Experimental Results

### 4.1. Simulation Results

**4.1.1. Line Fitting Simulation.** The principle of line fitting is as follows: assume that there exists a point set  $\{(x_1, y_1), (x_2, y_2), \dots, (x_{n-1}, y_{n-1}), (x_n, y_n)\}$ , the fitting line is expressed as  $ax + by + c = 0$ , and  $a, b, c$  satisfy

$$\sum_{i=0}^n \begin{bmatrix} x_i^2 & x_i y_i & x_i \\ x_i y_i & y_i^2 & y_i \\ x_i & y_i & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = 0. \quad (26)$$

We generated a random simulation data set which fitted with a given line. The standard deviation of inlier points was set to 1, and the outlier points obeying a uniform distribution in the minimum rectangular region contained the all inliers. The size of test points was set to 500. The outlier rates were set to 20%, 30%, 40%, 50%, 60%, 70%, and 80%, respectively. During the simulation, 2 points were used to estimate the line function. And  $\alpha, r_0$  were set to 0.05 and 2, respectively. We performed the RANSAC algorithm with fixed iterations and recorded the recall ratio in different outlier rates.

Figure 3 shows the average recall ratios of different RANSAC algorithms by recording 50 repeat experiments of the line fitting. In Figure 3, the recall ratio of the proposed IPGSAC algorithm is the largest in those five algorithms and it is nearly 100% when the outlier rate is lower than 60%. With the increasing of outlier rate, the recall ratios of other four RANSAC algorithms are rapidly decreased and the proposed algorithm still keeps the highest recall ratio with a small reduction. Table 1 illustrates the statistics of iterations when IPGSAC is convergent in 50 repeat experiments. The result illustrates that the average iteration of the proposed algorithm

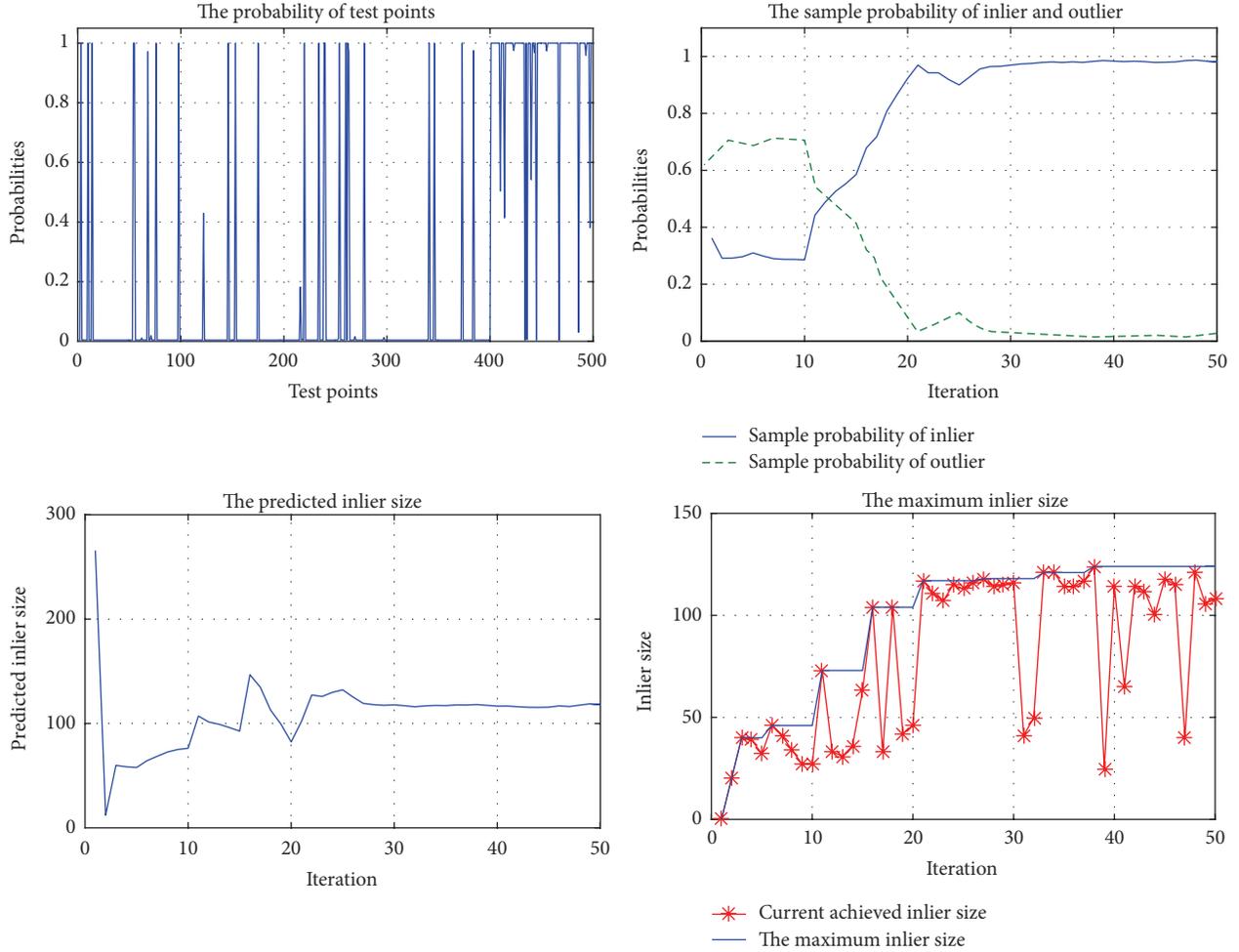


FIGURE 4: The simulation result of line fitting when the outlier rate is 80%.

TABLE 1: The statistics of iterations when IPGSAC algorithm is converged in line fitting.

Outlier rate	20%	30%	40%	50%	60%	70%	80%
Min iterations	3	3	3	4	3	7	12
Max iterations	19	17	17	46	35	76	92
Average iterations	7.95	8.65	8.60	13.05	16.80	27.05	41.55

is much smaller than the maximum number of iterations estimated by (1).

Figure 4 shows the estimated parameters when the outlier rate is 80% within 50 iterations. At the beginning of iteration, the sample probability of outlier points was much larger than the inliers. During the iteration, the probabilities of test points were updated by using proposed evaluation rule. After 20 iterations, the probabilities of inlier points were near to 1 and outlier was near to 0. The sample probability of inlier points was increased rapidly with the increasing of the probability of inlier points. Based on the principle of inverse mapping sampling strategy, the selection of test sample set trended to

the points with high probability in our proposed algorithm. It effectively improved the convergence rate of our algorithm. In fact, the algorithm reached the stop criterion in the 21st iteration. After 26 iterations, the estimated number of inlier set was converged to determined values, and it was high consistent with the actual acquired number of inlier points. Figure 5 illustrates the line fitting results when the outliers are 70% and 80%. The line fitting simulation results show that the proposed IPGSAC algorithm can rapidly localize inlier points. The probability updating algorithm can effectively improve the selection probability of inlier point and speed up the convergence rate of the proposed algorithm.

**4.1.2. Homography Estimation Simulation.** Assume that  $u_{\text{curr}} = \{u_0, u_1, \dots, u_n\}$  and  $u_{\text{ref}} = \{u_0^{\text{ref}}, u_1^{\text{ref}}, \dots, u_n^{\text{ref}}\}$  are two correspondent image feature point sets. The homography relationship is defined as

$$au_i = Hu_i^{\text{ref}}, \quad (27)$$

where  $a$  is a scaling factor and  $H$  is a homography matrix. According to *Lie* algebra, homography matrix  $H$  can be

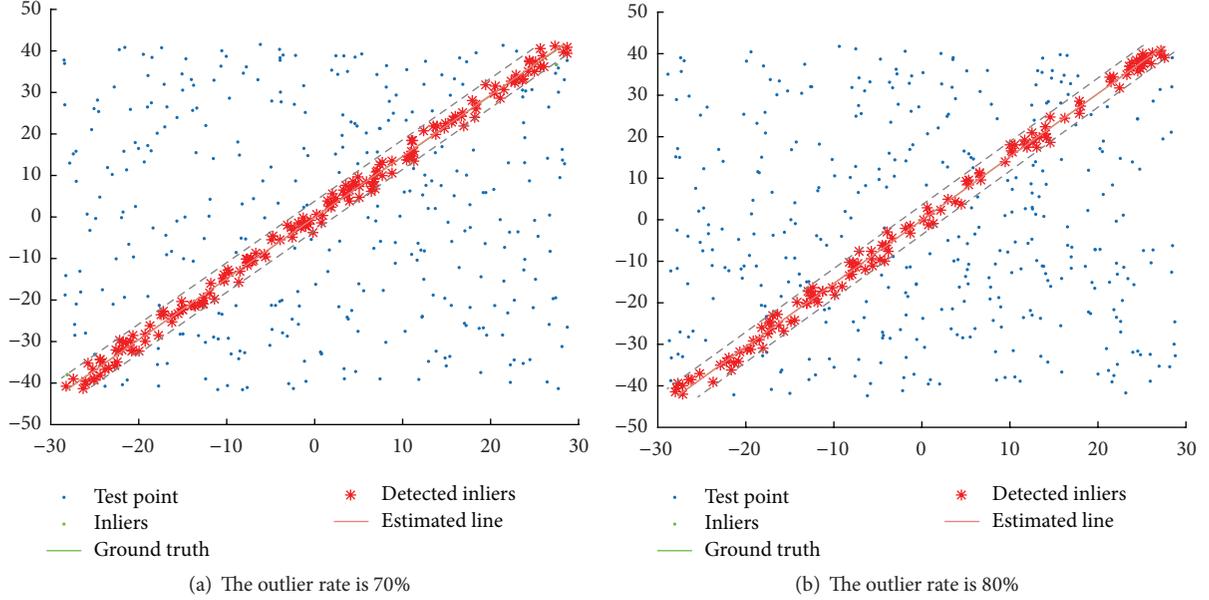


FIGURE 5: The line fitting results of the proposed algorithm when the outlier rates are 70% and 80%.

expressed as an exponent mapping by using an 8-variable  $x = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$ :

$$H = \exp(A(x)),$$

$$A = \begin{bmatrix} x_5 & x_3 & x_1 \\ x_4 & -x_5 - x_6 & x_2 \\ x_7 & x_8 & x_6 \end{bmatrix}. \quad (28)$$

In order to estimate the homography matrix  $H$ , an energy function is established by using the projection errors:

$$E = \sum_{i=0}^n \|r_i\|, \quad r_i = \pi(Hu_i^{\text{ref}}) - u_i, \quad (29)$$

where  $m$  ( $m \geq 4$ ) is the number of matching pairs which are used for the homography estimation. To optimize this object function, we follow the linear expansion and Newton iteration method. The minimum of energy function (29) is achieved by solving (30) iteratively:

$$\sum_{i=1}^n \left( \frac{\partial r_i}{\partial x} \right)^T \frac{\partial r_i}{\partial x} \xi = - \sum_{i=1}^n \left( \frac{\partial r_i}{\partial x} \right)^T r_i. \quad (30)$$

A random simulation data set was generated which fitted with a given homography relationship. To enhance reliability and authenticity of our simulation, the standard deviation of inlier points was set to 1 pixel and the outlier points obeyed the uniform distribution within the image. The size of test points was set to 1,000, and the outlier rates were set to different values which were the same as in the line fitting simulation. During the simulation, we selected four points to estimate the homography.  $\alpha, r_0$  were set as 0.05 and 2, respectively. The simulation was repeated 50 times to achieve a more reliable result in each outlier rate.

TABLE 2: The statistics of iterations when IPGSAC is converged in homography estimation simulation.

Outlier rate	20%	30%	40%	50%	60%	70%	80%
Min iterations	3	3	3	3	4	4	3
Max iterations	10	15	14	13	21	42	99
Average iterations	4.66	5.42	5.46	5.82	8.98	14.60	26.74

Figure 6 shows the compared results of the simulation. In the simulation, the recall ratio of the proposed algorithm is much higher than other four RANSAC algorithms in any tested outlier rates. Even when the outlier is 80%, this proposed algorithm is still available with the recall ratio 0.85962 after 20 iterations. Table 2 demonstrates the statistics of iterations when IPGSAC is converged in the homography estimation simulation. These results illustrate that the proposed IPGSAC algorithm is more effective than other four RANSAC algorithms and demonstrate the high performance of the proposed algorithm.

**4.1.3. Fundamental Matrix Estimation Simulation.** In this paper, 8-point method is used for fundamental matrix estimation. Assume that  $m = (u, v, 1)$  and  $m' = (u', v', 1)$  are an image matching pair; the fundamental matrix is limited with following equation:

$$m' F m = 0, \quad (31)$$

where  $F$  is a  $3 \times 3$  matrix. We follow the method in [6] to estimate  $F$ , and the error function is defined as

$$r = \frac{(m' F m)^2}{\|F^T m'\| + \|F m\|}. \quad (32)$$

The simulation data was generated similarly as in homography simulation which fitted with a given fundamental

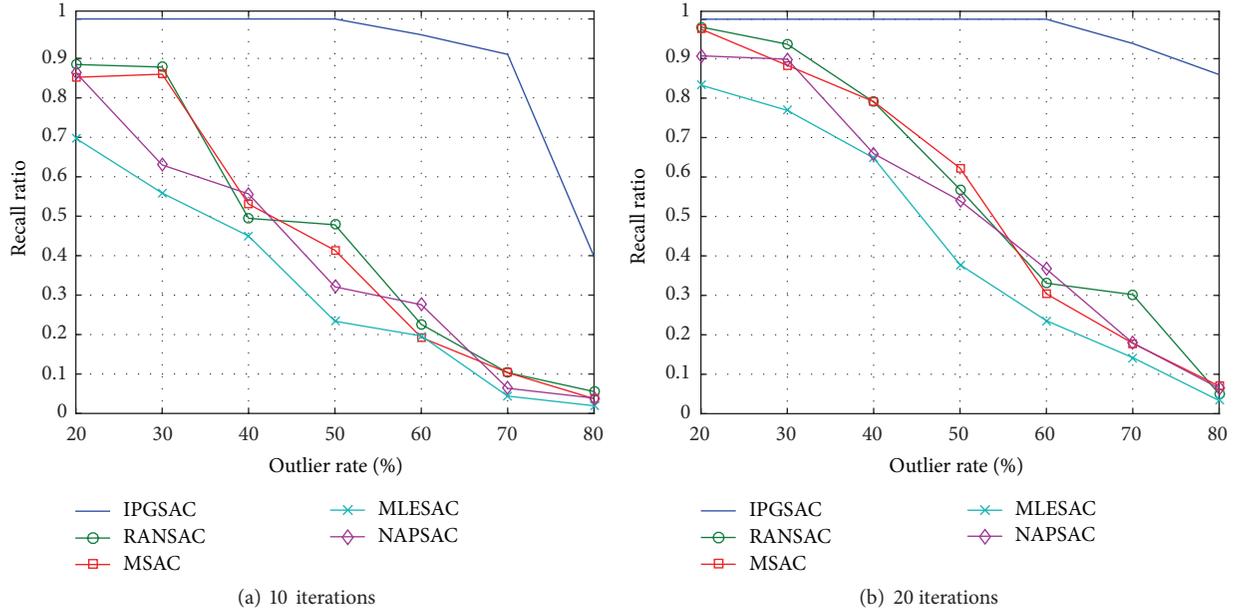


FIGURE 6: The recall ratios with 10 and 20 iterations in homography estimation simulation.

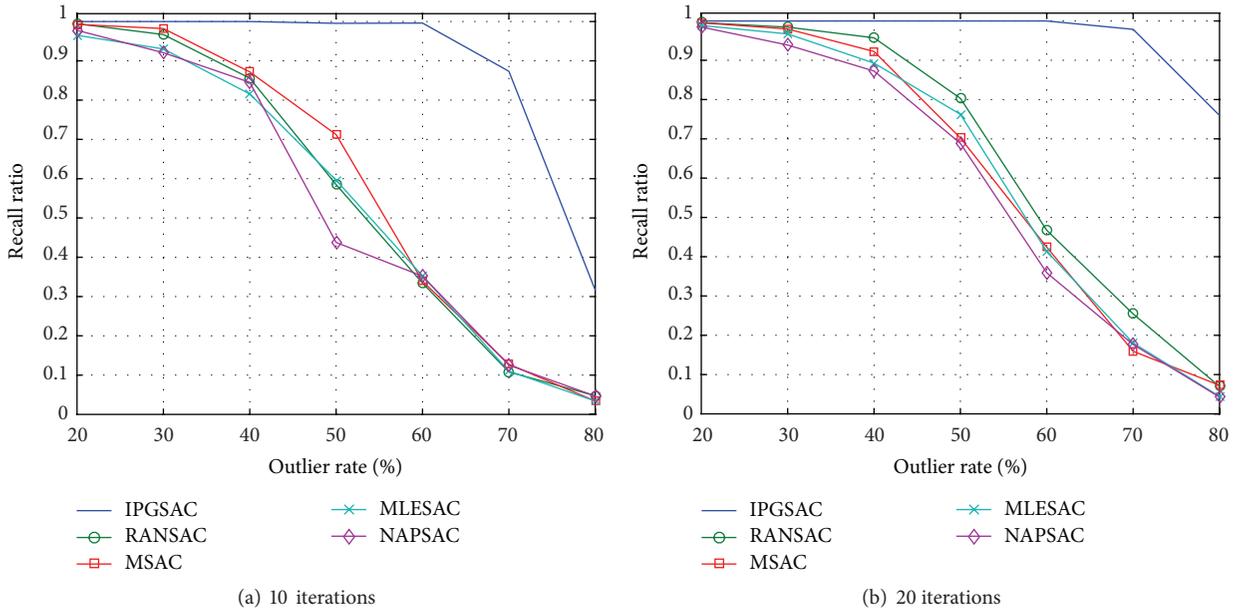


FIGURE 7: The recall ratios with 10 and 20 iterations in fundamental matrix estimation simulation.

relationship. The standard deviation of inlier points was set to 1 pixel and the outlier points obeyed the uniform distribution within the image. During the experiment,  $\alpha$  and  $r_0$  were set to 0.05 and 1, respectively. We also recorded the average recall ratios in 50 repeat simulations when the outlier rates were 20%, 30%, 40%, 50%, 60%, 70%, and 80%, respectively.

Figure 7 shows the recall ratios of different RANSAC algorithms in 10 and 20 iterations, respectively. Due to the low probability of inlier test set selection, the recall ratios of the other four RANSAC algorithms begin to decrease rapidly when the outlier rate is larger than 30%. As the fundamental

matrix estimation is solved with several linear equations, the constraint condition is much weaker than the homography relationship. The average recall ratios of fundamental matrix estimation are a little better than the homography estimation. Table 3 illustrates the statistics of iterations when IPGSAC is converged with 50 repeat experiments. It shows that it only needs 26.5 iterations in average to deal with the outlier rate 80%, and it is much more efficient than other four algorithms.

**4.1.4. The Time Cost and Accuracy of IPGSAC.** In the previous section, some results are given to illustrate the high

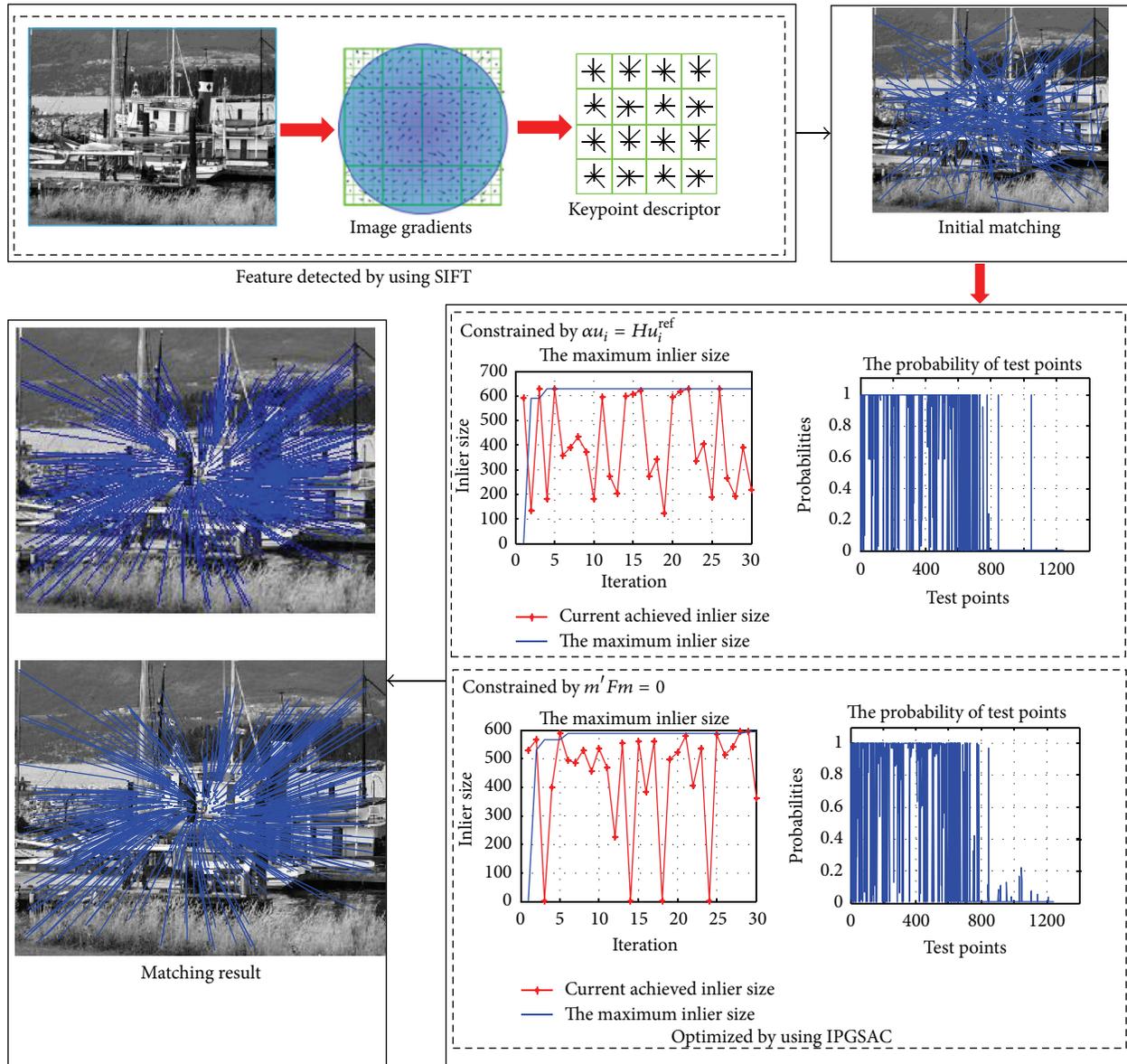


FIGURE 8: The image matching experimental results.

performance of the proposed algorithm. In some application of RANSAC, the time cost and the accuracy of the proposed algorithm are important issue for the real time requirement. As a typical application, the fundamental matrix estimation simulation was used for evaluating the time cost and the accuracy of the proposed algorithm. To ensure the convergence of each RANSAC algorithm, the maximum number of iterations was set to 100 and the other parameters were set the same as those in Section 4.1.3. All timing results were obtained on a PC with Intel i5 760 CPU (2.80 GHz) and 4 GB RAM.

The time cost, mean absolute error, and recall ratio of five RANSAC algorithms are illustrated in Tables 4–6. As shown in the tables, the recall ratios of those five RANSAC algorithms are very similar to each other when the outlier rate is lower than 40%. Generally, a lot of iterations are needed to ensure that an inlier set is found in the standard RANSAC,

MSAC, MLESAC, and NAPSAC. Therefore, the time cost of those algorithms is much higher than the proposed algorithm. The recall ratios of those algorithms are decreased rapidly with the outlier rate increasing. The time cost of the proposed algorithm is the lowest with the smallest mean absolute error and highest recall ratio during the simulation. And the time cost has a small increase with the raising of the outlier rate. The results further demonstrate the low time cost and the high accuracy of the proposed algorithm.

#### 4.2. Real Experimental Results

4.2.1. Image Matching Experiments. During the experiment, two images taken from different perspective in a real scene [34] were used for image matching. Firstly, the SIFT algorithm was used for feature detection and matching, as shown in Figure 8. There were 1244 feature pairs which were matched

TABLE 3: The statistics of iterations when IPGSAC is converged in fundamental estimation simulation.

Outlier rate	20%	30%	40%	50%	60%	70%	80%
Min iterations	3	3	3	3	3	3	3
Max iterations	7	6	9	8	39	54	99
Average iterations	3.60	3.94	4.02	4.22	10.88	10.88	26.5

TABLE 4: The time cost of different RANSAC algorithms.

Outlier rate	20%	30%	40%	50%	60%	70%	80%
IPGSAC	<b>0.0479</b>	<b>0.0493</b>	<b>0.0519</b>	<b>0.0509</b>	<b>0.0608</b>	<b>0.0628</b>	<b>0.0959</b>
RANSAC	0.4378	0.4376	0.4379	0.4439	0.4695	0.4415	0.429
MSAC	0.4299	0.4295	0.4296	0.4371	0.464	0.4332	0.4285
MLESAC	0.4505	0.4488	0.4472	0.4541	0.4786	0.4504	0.4455
NAPSAC	0.4374	0.4367	0.4364	0.444	0.4766	0.4421	0.4365

TABLE 5: The mean absolute error of different RANSAC algorithms.

Outlier rate	20%	30%	40%	50%	60%	70%	80%
IPGSAC	<b>0.2159</b>	<b>0.2347</b>	<b>0.2348</b>	<b>0.2672</b>	<b>0.2396</b>	<b>0.2533</b>	<b>0.2926</b>
RANSAC	0.3099	0.3506	0.3685	0.4256	0.5444	0.7716	0.8717
MSAC	0.256	0.29	0.327	0.4541	0.5019	0.7787	0.7886
MLESAC	0.2581	0.2824	0.3101	0.376	0.4236	0.5553	0.5569
NAPSAC	0.2821	0.3367	0.4036	0.4955	0.5591	0.776	0.7858

TABLE 6: The recall ratio of different RANSAC algorithms.

Outlier rate	20%	30%	40%	50%	60%	70%	80%
IPGSAC	<b>0.9988</b>	0.9973	<b>0.9951</b>	<b>0.9901</b>	<b>0.9763</b>	<b>0.9773</b>	<b>0.9494</b>
RANSAC	0.9988	<b>0.9974</b>	0.9941	0.9813	0.904	0.5588	0.1517
MSAC	0.9988	0.9973	0.9933	0.9884	0.9293	0.6061	0.1472
MLESAC	0.9983	0.9961	0.9884	0.9704	0.877	0.5245	0.1954
NAPSAC	0.9987	0.9967	0.992	0.9689	0.8657	0.5206	0.1525

by using the distance of feature descriptor. We used 8-point method and 4-point method for the fundamental matrix and homography matrix estimation, respectively. We also replicated all the matching experiment 50 times; the average outlier rate was about 52% which was achieved by using the proposed algorithm.

During the homography estimation experiment,  $\alpha$  and  $r_0$  were set to 0.05 and 4, respectively. The image matching result is shown in Figure 8. The probabilities of test points have obvious boundaries and the probability of inlier points is near to 1. Table 7 illustrates the number of inlier points achieved by different RANSAC algorithms with different iterations. The experimental result is highly consistent with the simulations. This proposed IPGSAC algorithm needs no more than 20 iterations to get convergence in average.

In fundamental matrix estimation,  $\alpha$  and  $r_0$  were set to 0.05 and 1, respectively. The probabilities of test points are quite similar with homography for foundation estimation. Table 8 shows the number of inlier points achieved by different RANSAC algorithms with different iterations. This experimental result is also similar with the simulation of fundamental matrix estimation. This proposed IPGSAC

TABLE 7: The inlier size with different iterations in homography estimation.

Iterations	IPGSAC	RANSAC	MSAC	MLESAC	NAPSAC
10	<b>593.46</b>	101.7	129.48	71.40	138.26
20	<b>595.46</b>	187.94	242.42	118.80	261.42
30	<b>596.46</b>	301.68	290.10	194.66	299.56

TABLE 8: The inlier size with different iterations in fundament estimation.

Iterations	IPGSAC	RANSAC	MSAC	MLESAC	NAPSAC
10	<b>595.36</b>	311.16	311.38	356.28	288.34
20	<b>610.80</b>	386.48	389.62	462.12	401.40
30	<b>611.30</b>	446.10	427.58	511.04	445.52

algorithm also needs no more than 20 iterations to achieve a convergent solution in average. While the homography matrix is a plane constraint of features and the fundamental matrix is a common constraint in image matching, the result has some tiny difference with homography estimation. All

TABLE 9: The evaluation of absolute translational error ( $m$ ).

		ICP-based method	Dense image-based method	Our method
Freiburg1/desk	min	0.092303	0.016672	<b>0.007411</b>
	max	1.078730	<b>0.178265</b>	0.246963
	RMSE	0.5715	0.1134	<b>0.093726</b>
Freiburg1/desk2	min	0.016459	0.021073	<b>0.003560</b>
	max	1.024154	0.427586	<b>0.317763</b>
	RMSE	0.368354	0.119614	<b>0.096128</b>



(a) Mobile robot platform



(b) Experimental environment

FIGURE 9: The platform and experimental environment.

those model estimation results demonstrate the feasibility and effectiveness of the proposed IPGSAC algorithm.

**4.2.2. The 3D Map Building Experiment.** The 3D mapping experiment was taken in an indoor environment shown in Figure 9. The experimental platform consisted of a Pioneer3-DX mobile robot and a Microsoft Kinect. The P3-DX mobile robot is an intelligent mobile platform equipped with an embedded PC. The Kinect provides the depth and color data captured in the experimental scene. The maximum field of view of Kinect is  $52^\circ$  and the image resolution is  $640 \times 480$  with the refresh frequency 30 Hz. In the experiment, the Kinect was fixed on the mobile robot and captured the depth and color data in the scene. The distance threshold  $d_0$  and angle threshold  $\theta_0$  were set to 0.2 and 0.3, respectively. The tolerance threshold  $r_0$  was equal to 1. The pseudo minimum inlier rate  $\alpha$  was 0.05. To localize the robot, four points were used in the IPGSAC-based pose estimation procedure.

To evaluate the proposed algorithm, we compare our method with standard ICP-based tracking algorithm and a dense image-based tracking algorithm [9] using the RGBD SLAM benchmark provided by Technical University of Munich [35]. Table 9 shows the evaluations of the absolute translational error by using those three algorithms with two typical indoor data sets. In the experiment, ICP-based tracking method almost failed with the largest evaluation of the absolute translational error. Due to the blur and occlusion regions of the input data, the dense image-based method

easily falls into local optimum. In our proposed method, the data association was established by using the feature matching procedure and it sometimes noised by the mismatched features. And the noise would be removed in our IGPSAC-based pose estimation procedure. Although the IGPSAC-based pose estimation procedure is limited by the accuracy of feature detection, the smallest RMSE (Root Mean Square Error) of the absolute translational error is achieved by using our proposed method.

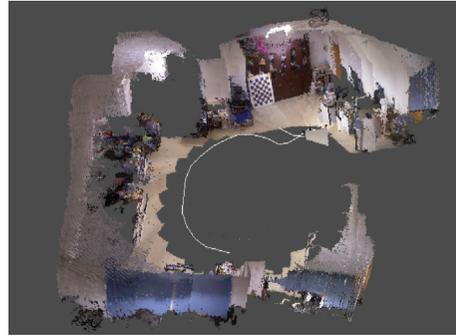
In the indoor experiment, the robot was instructed to traverse around in the environment. And the translational speed of mobile robot was about 0.1 m/s. The origin of coordinate of robot was set at the location where robot started to move. While the mobile robot was moving, the depth and color image were captured by the Kinect and the robot's pose was estimated by using the algorithm detailed in Section 3. When a keyframe was captured, the global map was updated with the new coming keyframe and the loop detection procedure was performed to estimate the relation pose between this new coming keyframe and previous nonadjacent keyframes. When a loop closure was detected, the TORO optimization algorithm would be performed to achieve a global optimization of generated map. Table 10 illustrates the number of iterations when the proposed IPGSAC is converged during the mapping experiment. According to (1), the minimum iteration is in the range of 60~7725 when the confidence level is 98%. The iteration when the proposed IPGSAC-based algorithm is converged is much littler than the estimated

TABLE 10: The statistics of iterations when IPGSAC is converged in 3D map building.

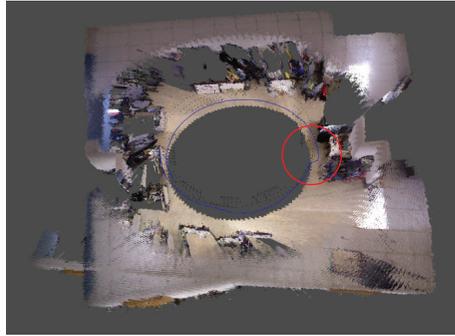
Outlier rate	0.25–0.35	0.35–0.45	0.45–0.55	0.55–0.65	0.65–0.75	075–0.85
Count	19	31	247	354	112	19
Min iterations	2	2	2	2	2	7
Max iterations	9	25	46	92	61	87
Average iterations	5.21	7.70	9.06	13.66	15.47	20.29



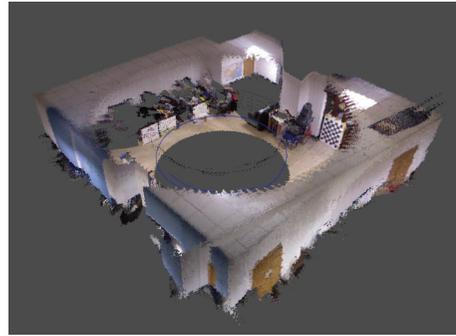
(a) ICP-based method



(b) Dense image-based method



(c) RANSAC-based method



(d) The proposed IPGSAC-based method

FIGURE 10: The generated mapping before the loop optimization.

minimum iterations. It is highly consistent with our previous simulations and verifies the high efficiency of the proposed algorithm.

Moreover, the ICP-based pose estimation, the dense image-based pose estimation, and the standard RANSAC-based pose estimation were also employed for map building. Figure 10 shows the generated mapping before the loop optimization by using those methods. During the map building, the ICP-based tracking algorithm failed due to the similar spatial structure of the input point clouds. Although the dense image-based method tracked the mobile robot well, the generated map sank seriously. In the RANSAC-based method, it was sometimes easy to converge to local optima and tracking procedure would fail as indicated by the red circle in Figure 10(c). In the IPGSAC-based pose estimation method, the pose estimation procedure worked well and the generated map had a very slightly subsidence during the experiment. With those estimated poses, the loop closure was easily detected and it was much easier to perform loop optimization procedure and achieve a global consistent map. The final mapping experimental result of our method is shown in

Figure 11. All the experimental results demonstrate the feasibility and effectiveness of this proposed algorithm.

## 5. Conclusions

In this paper, we present a novel RANSAC algorithm for robot 3D map building. Firstly, two types of the probabilities are updated and evaluated for test points by using a total probability formula and the maximum number of inlier points. Then, a more reliable evaluation of test points is achieved by using the DS evidence theory. To reduce the influence of incorrect judgment, the theories of forgetting are utilized to filter out the unstable inlier points. Finally, an inverse mapping sample algorithm is adopted to sample the points based on the updated probabilities of points. All simulation and real experimental results show that the proposed algorithm can quickly and efficiently achieve a stable convergence in a few iterations and demonstrate the feasibility and effectiveness of the proposed algorithm.

As is illustrated in the experiment results, the proposed algorithm is much more efficient than some other RANSAC

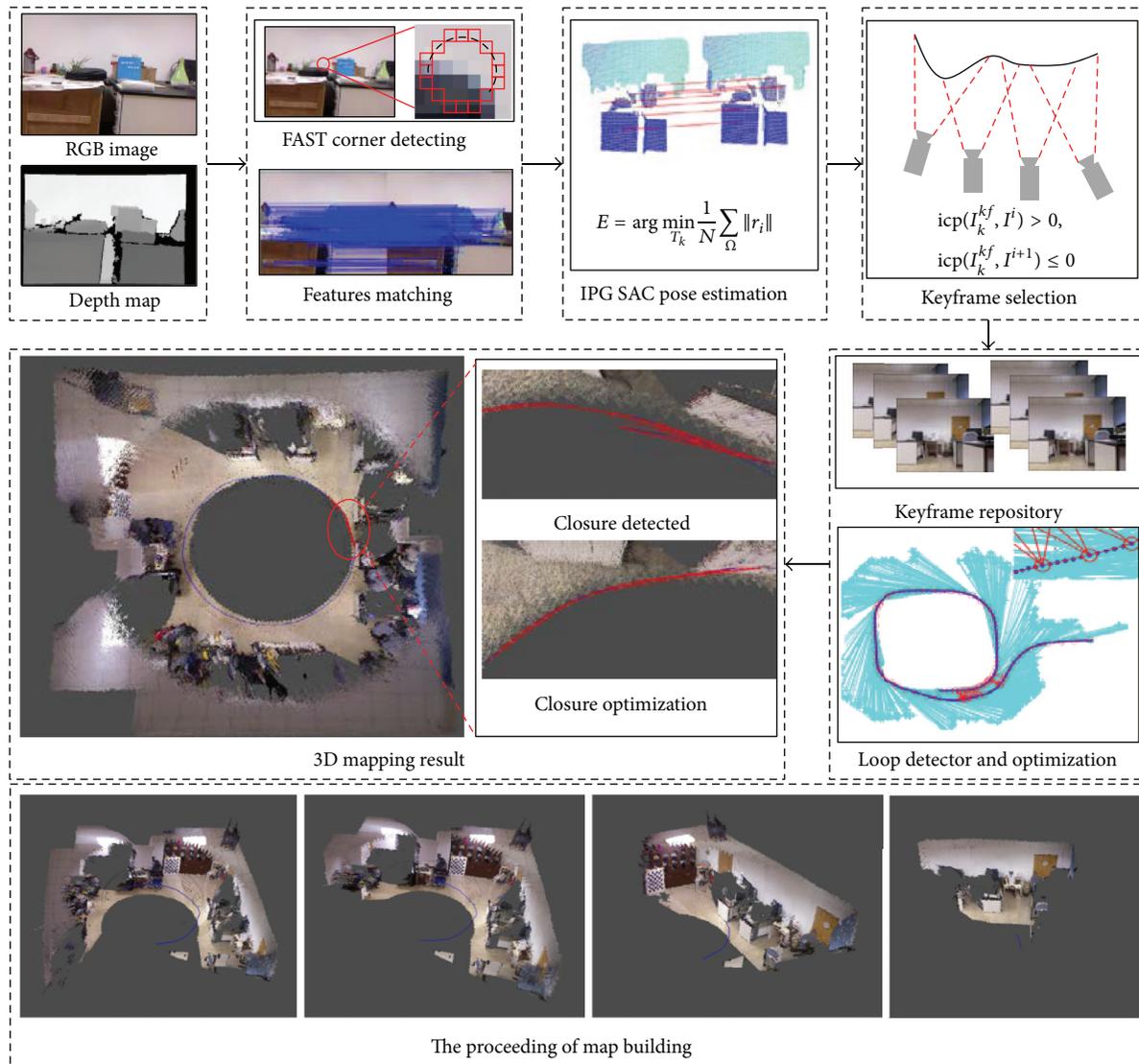


FIGURE 11: The 3D map building experimental result.

algorithms when the outlier rate is lower than 80%. In fact, the utmost outlier rate of this proposed algorithm is 85%. When the outlier rate rises to 90%, this proposed algorithm becomes unstable and unviable. In some extreme cases, it needs more than 2,000 iterations to achieve a correct solution. In future work, we will further improve the robustness and efficiency of our proposed algorithm to achieve a stable solution for the higher outlier rate. And an adapt threshold scheme will be proposed to enhance the robust of our method.

### Competing Interests

The authors declare that they have no competing interests.

### Acknowledgments

The research work is financially supported by Promotion Project on Intelligent Robot in Beijing University of

Technology “Key Technologies on Modularization and Practicalization of Intelligent Service Robot” and the Natural Science Foundation of China (61175087).

### References

- [1] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” in *Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '07)*, pp. 225–234, IEEE, Nara, Japan, November 2007.
- [2] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “RGB-D mapping: using Kinect-style depth cameras for dense 3D modeling of indoor environments,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.
- [3] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, “An evaluation of the RGB-D SLAM system,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '12)*, pp. 1691–1696, IEEE, St. Paul, Minn, USA, May 2012.

- [4] B. K. Horn and B. G. Schunck, "Determining optical flow," in *Proceedings of the Technical Symposium East*, pp. 319–331, International Society for Optics and Photonics, 1981.
- [5] R. Ranftl, S. Gehrig, T. Pock, and H. Bischof, "Pushing the limits of stereo using variational stereo estimation," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '12)*, pp. 401–407, Alcalá de Henares, Spain, June 2012.
- [6] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2nd edition, 2004.
- [7] X. Armangué and J. Salvi, "Overall view regarding fundamental matrix estimation," *Image and Vision Computing*, vol. 21, no. 2, pp. 205–220, 2003.
- [8] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '13)*, pp. 2100–2106, IEEE, Tokyo, Japan, November 2013.
- [9] F. Steinbrücker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV '11)*, pp. 719–722, IEEE, Barcelona, Spain, November 2011.
- [10] P. H. S. Torr and A. Zisserman, "MLESAAC: a new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.
- [11] P. J. Rousseeuw, "Least median of squares regression," *Journal of the American Statistical Association*, vol. 79, no. 388, pp. 871–880, 1984.
- [12] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [13] P. H. S. Torr, "Bayesian model estimation and selection for epipolar geometry and generic manifold fitting," *International Journal of Computer Vision*, vol. 50, no. 1, pp. 35–61, 2002.
- [14] S. Obdrzalek, O. Chum, and J. Matas, "Enhancing RANSAC by generalized model optimization," in *Proceedings of the Asian Conference on Computer Vision (ACCV '04)*, 2004.
- [15] B. J. Tordoff and D. W. Murray, "Guided-MLESAC: faster image transform estimation by using matching priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1523–1535, 2005.
- [16] O. Chum and J. Matas, "Matching with proSAC—progressive sample consensus," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 220–226, IEEE, June 2005.
- [17] D. Myatt, P. Torr, S. Nasuto, J. Bishop, and R. Craddock, "NAPSAC: high noise, high dimensional robust estimation—it's in the bag," in *Proceedings of the 13th British Machine Vision Conference (BMVC '02)*, pp. 44.1–44.10, BMVA Press, Cardiff, Wales, 2002.
- [18] O. Chum and J. Matas, "Randomized ransac with  $T_{d,d}$  test," in *Proceedings of the British Machine Vision Conference*, vol. 2, pp. 448–457, September 2002.
- [19] J. Matas and O. Chum, "Randomized RANSAC with sequential probability ratio test," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 2, pp. 1727–1732, IEEE, Beijing, China, October 2005.
- [20] A. Hast, J. Nysjö, and A. Marchetti, "Optimal RANSAC—towards a repeatable algorithm for finding the optimal set," *Journal of WSCG*, vol. 21, no. 1, pp. 21–30, 2013.
- [21] V. Rodehorst and O. Hellwich, "Genetic algorithm sample consensus (gasac)—a parallel strategy for robust parameter estimation," in *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '06)*, p. 103, IEEE, June 2006.
- [22] M. T. El-Melegy, "Model-wise and point-wise random sample consensus for robust regression and outlier detection," *Neural Networks*, vol. 59, pp. 23–35, 2014.
- [23] G. Shafer, *A Mathematical Theory of Evidence*, vol. 1, Princeton University Press, Princeton, NJ, USA, 1976.
- [24] T. Botterill, S. Mills, and R. Green, "New conditional sampling strategies for speeded-up ransac," in *Proceedings of the 13th British Machine Vision Conference (BMVC '09)*, pp. 1–11, September 2009.
- [25] J. D. Hol, T. B. Schön, and F. Gustafsson, "On resampling algorithms for particle filters," in *Proceedings of the Nonlinear Statistical Signal Processing Workshop (NSSPW '06)*, pp. 79–82, IEEE, Cambridge, UK, September 2006.
- [26] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings F—Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.
- [27] J. Smisek, M. Jancosek, and T. Pajdla, "3D with kinect," in *Consumer Depth Cameras for Computer Vision*, pp. 3–25, Springer, 2013.
- [28] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006. Proceedings, Part I*, vol. 3951 of *Lecture Notes in Computer Science*, pp. 430–443, Springer, Berlin, Germany, 2006.
- [29] E. Rosten, R. Porter, and T. Drummond, "Faster and better: a machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2010.
- [30] M. Calonder, V. Lepetit, M. Özuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: computing a local binary descriptor very fast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, 2012.
- [31] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Robotics-DL Tentative*, pp. 586–606, International Society for Optics and Photonics, 1992.
- [32] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard, "Efficient estimation of accurate maximum likelihood maps in 3D," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 3472–3478, IEEE, San Diego, Calif, USA, November 2007.
- [33] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A tree parameterization for efficiently computing maximum likelihood maps using gradient descent," in *Robotics: Science and Systems*, 2007.
- [34] Visual Geometry Group, "Oxford visual geometry group affine covariant regions datasets," 2005, <http://www.robots.ox.ac.uk/~vgg/data/data-aff.html>.
- [35] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proceedings of the 25th IEEE/RSJ International Conference on Robotics and Intelligent Systems (IROS '12)*, pp. 573–580, IEEE, Vilamoura, Portugal, October 2012.

## Research Article

# An Analytical Measuring Rectification Algorithm of Monocular Systems in Dynamic Environment

**Deshi Li<sup>1,2</sup> and Xiaoliang Wang<sup>1</sup>**

<sup>1</sup>*Electronic Information School, Wuhan University, Wuhan, Hubei 430072, China*

<sup>2</sup>*Collaborative Innovation Center of Geospatial Technology, 129 Luoyu Road, Wuhan 430079, China*

Correspondence should be addressed to Xiaoliang Wang; [xiaoliangwang@whu.edu.cn](mailto:xiaoliangwang@whu.edu.cn)

Received 16 October 2015; Revised 25 January 2016; Accepted 2 March 2016

Academic Editor: Yassine Ruichek

Copyright © 2016 D. Li and X. Wang. 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.

Range estimation is crucial for maintaining a safe distance, in particular for vision navigation and localization. Monocular autonomous vehicles are appropriate for outdoor environment due to their mobility and operability. However, accurate range estimation using vision system is challenging because of the nonholonomic dynamics and susceptibility of vehicles. In this paper, a measuring rectification algorithm for range estimation under shaking conditions is designed. The proposed method focuses on how to estimate range using monocular vision when a shake occurs and the algorithm only requires the pose variations of the camera to be acquired. Simultaneously, it solves the problem of how to assimilate results from different kinds of sensors. To eliminate measuring errors by shakes, we establish a pose-range variation model. Afterwards, the algebraic relation between distance increment and a camera's poses variation is formulated. The pose variations are presented in the form of roll, pitch, and yaw angle changes to evaluate the pixel coordinate incensement. To demonstrate the superiority of our proposed algorithm, the approach is validated in a laboratory environment using Pioneer 3-DX robots. The experimental results demonstrate that the proposed approach improves in the range accuracy significantly.

## 1. Introduction

The applications of mobile robots for observation and rescue missions have received an increasing attention in recent years. Current advances in sensing and computing promote mobile robots as a suitable option in occasions such as search and rescue, SLAM (Simultaneous Localization and Mapping), automatic navigation, and target detection. For mobile robots, retrieving their position is one of the important issues. In recent years, to solve this problem, vision sensors have attracted a lot of attention because vision sensors are relatively inexpensive and compact with low power consumption. Furthermore, methods using vision sensors can localize the robot in various environments where it is difficult for general localization methods like wheel odometry and GPS. If localization can be performed only using image information, a robot's flexibility will be improved remarkably.

However, the precondition to a successful intelligent robot system is the exact perception of surroundings, where the range and azimuth information of targets around play an

important role. The range estimation algorithms using vision sensors are known as VO (visual odometry).

Approaches for range estimation can mainly be divided into three categories: radar-based, laser-based, and vision-based. As a typical paradigm of noncontact approaches, ultrasonic sensors have the advantages of time efficiency and measurement accuracy. However, it is arduous to detect those objects with small surfaces or situated at a wide angle related to the ultrasonic sensor(s). Among all perception sensors, computer visions have an added advantage of acquiring large amount of information at a lower cost. The vision-based method can solve both range and azimuth estimation problems using only the acquired image themselves. There has been much interest in research on object detection by stereo camera [1–4], but the monocular camera is still strongly advantageous for its large sensing area, low cost, and easy installation.

To achieve precise VO in outdoor environments, some problems remain to be solved. In this work, the problems of VO on bumpy courses which exist mostly in outdoor

environments are considered, where the bumpy courses mean that the environments include rough roads on which the VO accuracy is dynamically affected by the pose change of vision sensors. Furthermore, if precise VO is realized in the environments including rough roads, we believe that it can be utilized in any outdoor environments.

In the researches of intelligent unmanned vehicle systems, computer vision generally adopts the methods of imaging processing algorithms. In those works, the image features are extracted, along with the model of the ambient environment, for vehicle localization and obstacle avoidance. Range and azimuth information are then refined from the above model using vision system. It is unrealistic to assume that the road is absolutely flat in the process. This paper concentrates on the dynamic measurement rectification problem in which the camera pose changes abruptly. This approach is particularly suitable for applications such as navigating the autonomous vehicles running on rough terrains. Pose variations are firstly measured by a three-axis angle sensor and sequentially applied to calculate the distance offsets using the proposed range-pixel model. Although monocular visual odometry has an advantage in wide FOV (field of view), factors such as lighting conditions, shadows, and random noise would unavoidably decrease the measurement precision, which are induced from both human limitations and sensor characteristics. On the contrary, noncontact sensors such as sonar are typically not susceptible to those external conditions which would infect the result accuracy. Nevertheless, one main defect is the existence of inherent blind areas. In the view of these possible advantages and corresponding limitations, sensor assimilation technique based on OI (Optimal Interpolation) method is employed. The main contributions of this paper are summarized as follows.

- (1) The relation between range increment and camera's pose variation has been formulated, based on which a feasible data rectification algorithm has been designed to modify the metrical results. To the best of our knowledge, it is the first work to solve range estimation problem under camera shaking conditions.
- (2) An improved estimation mechanism of range information in OI model has been developed, which enhances adaptability and accuracy of multisensor measuring system.
- (3) Experiments on mobile robots and analytical results have been demonstrated.

The rest of this paper is organized as follows. The following section will provide some background and a more detailed literature review. Section 3 defines the problems and related literatures. Section 4 details the proposed approach for measurement rectification and sensor fusion. Finally, experiment results and conclusions are given in Sections 5 and 6.

## 2. Related Works

Visual distance estimation is a specialized set of approaches which focus on real-time and accurate image capture

followed by range information acquirement. Several of these mechanisms have been developed as foundational elements of 3D reconstruction, simultaneous localization, and map building.

Some basic algorithms as well as their improvements for range estimation have been developed: epipolar constraint model [2], defocusing method [3–5], coordinate mapping scheme [6, 7], and camera movement approach [1, 8]. Katsuyuki et al. proposed a coupled estimation of unknown vehicle width and following distance by sequential Bayesian estimation. The method can run in real-time and produce highly accurate estimation of the following distance under a precondition that no camera shaking happens.

Those proposed methods can be divided into two categories: monocular and stereo system. Monocular approaches involve a single no sophisticated camera that compute the pixel size or coordinates which are used for range estimation. Examples of these are studied in [9]. Stereo vision approaches can provide much higher accuracy than monocular, but they have small field of view and high operational complexity. Several intelligent and operable algorithms [10, 11] fall into this category.

Monocular and stereo vision approaches have advantages in different aspects. Monocular approaches are usually easy to be implemented and have optimal view scope. Meanwhile, they require much lower cost compared to the former. Stereo vision methods, in contrast, have a good performance in accuracy due to the subpixel synthetical localization technology, while their biggest drawback lies in the complicated operations and high computational complexity, especially during the calibration process.

Among these emerged researches, most work assumes the camera pose is fixed [3, 6, 9, 10, 12–14]. Some notable exceptions, which have similarity to the present work, are as follows. (1) Guo et al. [15] put forward a parallel constraint method based on the two lane boundaries. (2) Vehicles are equipped with an angle sensor to accurately acquire the pitching angle of the camera in [13, 16, 17], where the author proposed an improved algorithm in angle calculation by using a function of the angles representing the two parallel lane lines.

Some other approaches have also been proposed. Typical paradigms are as follows. Han et al. [18] devise a feature point based method for monocular measurement, but they hinder the real-time implementation. Malis and Rives [19] design a hybrid algorithm to minimize the token relative displacements between two frames and then estimate the image-space distance.

## 3. Problem Formulation

In Figure 1 suppose  $\mathbf{P}'$  is a point in the image plane of the camera in a pose of  $\mathbf{P}_s$ , and suppose that we have an estimate of the pose of the camera by a three-axis gyroscope. From this information a standard ground-constrained model [18] can be used to estimate the position of  $\mathbf{P}$  in the world coordinate. If the camera's pose suddenly changes to  $\mathbf{P}_f$ , we can use this information to project point  $\mathbf{P}$  into the camera's image plane,

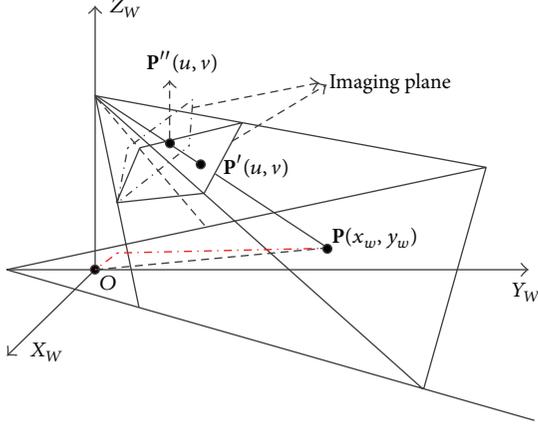


FIGURE 1: The imaging geometry of the observation model.

which obtains a second point  $P''$ . Now assuming that the pose measurement is reasonably accurate and that the position estimate algorithm works well, the problem is to estimate  $P$  utilizing measurements including  $P'$ ,  $P''$ , and pose variation of the camera.

The initial and final poses of the camera are denoted by  $P_s = (r_s, p_s, y_s)$  and  $P_f = (r_f, p_f, y_f)$ , respectively, where  $r_i$ ,  $p_i$ , and  $y_i$  ( $i = s, f$ ) stand for the initial roll, pitch, and yaw angles. Although the actual relative distance from the optical center to the target changes slightly, the measured results deviate from the truth significantly. This is mainly because of the nonlinear mapping between pixel coordinates and corresponding distance values. The problem is to correct the actual measurements to be close to the truth by eliminating the pose perturbation of the camera.

#### 4. Data Rectification Algorithm

In this section we describe our approach to the problem of monocular vision-based measurement rectification. Since a robot's trajectory is most conveniently described in a world coordinate system, while the target on the ground is generally described by its camera coordinate system, we start with a preview of these two coordinate systems. To model the problem in a general geometrodynamical architecture, the algebraic relation between the camera pose displacement and the displacement of measuring distance is derived.

**4.1. World and Camera Coordinates System.** Assume that  $XYZ$  and  $X'Y'Z'$  are, respectively, a world coordinate system and the camera's coordinate system as shown in Figure 2. The coordinates of a point  $P$  under these two coordinate systems are transformed by

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{R} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} + \mathbf{T}, \quad (1)$$

where  $(X, Y, Z)$  and  $(X', Y', Z')$  are point  $P$  coordinates in the world and robot camera coordinate system. Moreover,

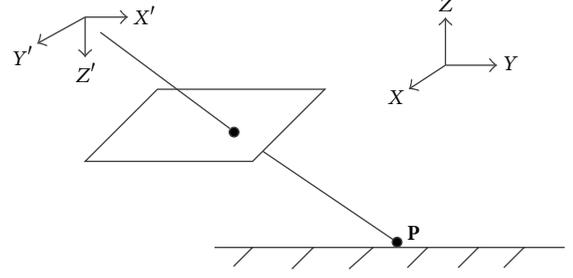


FIGURE 2: The world and camera coordinate systems.

$\mathbf{R}$  and  $\mathbf{T}$  are, respectively, the rotation and translation from the camera to the world's coordinate system, which determine the position and orientation of the camera in the world coordinate system. Furthermore, for a 3D point  $P$  in the FOV of the camera, its image coordinates are given by the projection equation as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix}^T = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix}^T \cdot f, \quad (2)$$

where  $(u, v)$  are the coordinates of  $P$  in the image coordinate system and  $f$  is the camera's focal length.

**4.2. Chebyshev Best Uniform Approximation Rectification Algorithm.** The distance-orientation information between targets and the camera can be derived from corresponding pixel coordinate in the image [20, 21]. It is found that the ratio of image pixel motion to the camera rotation angles varies nonlinearly along the main optical axis. The main idea of the designed algorithm is to piecewise linearize the nonlinear rate and then calculate the rate of change with respect to rotation angles as well as the measured distance. Equation (3) presents the rotation matrix in 3D space:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} \cos \psi & 0 & -\sin \psi \\ 0 & 1 & 0 \\ \sin \psi & 0 & \cos \psi \end{bmatrix} \cdot \begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

Variations of pixel coordinates are associated with the world coordinates by a rotation matrix whose parameters are attitude angles of the camera, which is described by

$$\begin{aligned} \Delta u_{k+1} &= F_x \cdot \left( \frac{R_{11}^{k+1} x_w + R_{12}^{k+1} y_w}{R_{31}^{k+1} x_w + R_{32}^{k+1} y_w} - \frac{R_{11}^k x_w + R_{12}^k y_w}{R_{31}^k x_w + R_{32}^k y_w} \right), \\ \Delta v_{k+1} &= F_y \cdot \left( \frac{R_{21}^{k+1} x_w + R_{22}^{k+1} y_w}{R_{31}^{k+1} x_w + R_{32}^{k+1} y_w} - \frac{R_{21}^k x_w + R_{22}^k y_w}{R_{31}^k x_w + R_{32}^k y_w} \right), \end{aligned} \quad (4)$$

where inner parameters  $F_x$  and  $F_y$  are only determined by the CCD structure itself.

For the convenience of discussion, we assume that the camera poses change mainly along the yaw angle direction.

Denote  $\Psi(\theta, \psi, \varphi) = (R_{11}x_w + R_{12}y_w)/(R_{31}x_w + R_{32}y_w)$ ; using (3) and (4), we obtain

$$\Psi(\theta, \psi, \varphi) = \cos \partial + \sin \partial \cot \beta, \quad (5)$$

where

$$\begin{aligned} \partial &= \arctan\left(\frac{x_w}{y_w}\right) + \arctan\left(\frac{y_w}{x_w}\right), \\ \beta &= \varphi - \arctan\left(\frac{y_w}{x_w}\right). \end{aligned} \quad (6)$$

Substituting (3) in (5) results in

$$\Psi(\theta, \psi, \varphi) = \cot\left(\varphi - \arctan\left(\frac{y_w}{x_w}\right)\right). \quad (7)$$

From Figure 3, we can see that the slope of curves tends to be constant within a sliding interval of independent variable. This interval becomes smaller when the ratio of  $x_w$  to  $y_w$  increases. The Chebyshev approximation method has the characteristics of uniform approximation on selected closed-interval. Inspired by this, the nonlinear rate can be approximated by linear polynomial and the deviation caused by poses change of a camera can be effectively compensated. The second derivative is taken as

$$\begin{aligned} \Psi''(\varphi) &= 2 \cot\left(\varphi - \arctan\left(\frac{y_w}{x_w}\right)\right) \\ &\cdot \left(\cot\left(\varphi - \arctan\left(\frac{y_w}{x_w}\right)\right)^2 + 1\right). \end{aligned} \quad (8)$$

Considering the yaw angle variations of a PTZ (Pan/Tilt/Zoom) camera caused by uneven pavement during practical robot motion, a closed subinterval  $[a, b]$  ( $[a, b] \subseteq [0 \sim \pi/12]$ ) is chosen for further deduction. Since (8) is a continuous function and keeps consistency in sign, the best consistent approximation method can be used.

Denote  $a_1^* = (\Psi(b) - \Psi(a))/(b - a)$ , using this to acquire normal equation of approximation:

$$\Psi'(\varphi) = a_1^*. \quad (9)$$

Set the solution of (9) as  $x_\Delta$ . Then the approximation equation is written as follows:

$$p^*(x) = \frac{f(a) + f(x_\Delta)}{2} + a_1^* \cdot \left(x - \frac{a + x_\Delta}{2}\right). \quad (10)$$

We explore the slope of line after linear approximation to study the function of different ratios of  $x_w$  to  $y_w$ . Results show that the slope converges to its limit uniformly. Moreover, this constant value is irrelevant to the ratio above:

$$\kappa = \frac{\Psi(b) - \Psi(a)}{b - a}. \quad (11)$$

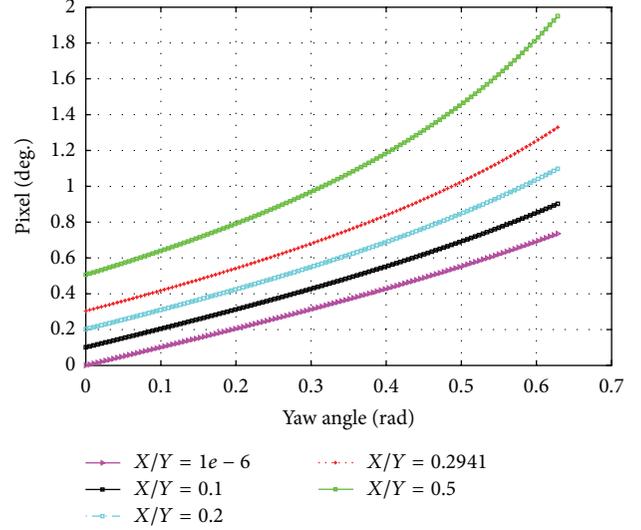


FIGURE 3: Pixel coordinates along  $U$ -axis vary with the yaw angles: a set of  $y_w/x_w$  values are considered.  $(x_w, y_w)$  is the coordinate of  $P$  in the world coordinate system.

Substituting (7) in (11) results in

$$\begin{aligned} \kappa &= \frac{1}{b - a} \cdot \frac{1 + (y/x)^2}{(\tan(b) - y/x)(\tan(b) - y/x)} \\ &\cdot (\tan(a) - \tan(b)). \end{aligned} \quad (12)$$

To demonstrate the convergence of  $\kappa$ , we have also analyzed the limit value given by

$$\lim_{y/x \rightarrow \infty} \kappa = \frac{\tan(a) - \tan(b)}{b - a}. \quad (13)$$

Solid curves in Figure 4 are the results of actual slope and linear approximation, respectively. These two curves coincide with each other well after a translation operation. This indicates a high accuracy in slope using linear approximation. Figure 5 manifests the convergence of slope related to a metric of  $y_w/x_w$ , which is in good agreement with experimental results. Another important property that should be noted is that the function value rapidly reaches convergence after a dramatic increase; that is, the measured range would vary with the metric  $y_w/x_w$  with high nonlinearity. This also implies that the measurement should be conducted on the smooth interval to reduce the errors caused by camera shakings. On the other hand, it is impossible to compensate the deviations when the metric is too small.

**4.3. Sonar and Camera Data Assimilation Model.** The Optimal Interpolation Algorithm is derived to generate the least squares results for vectors of observations and background fields assuming “a priori” known statistical models for the background error covariance. The Optimal Interpolation Technique, based on the minimization of variance estimation, plays an important role in data assimilation. It uses several different real-world observations to produce a corrected output, which is closer to the truth.

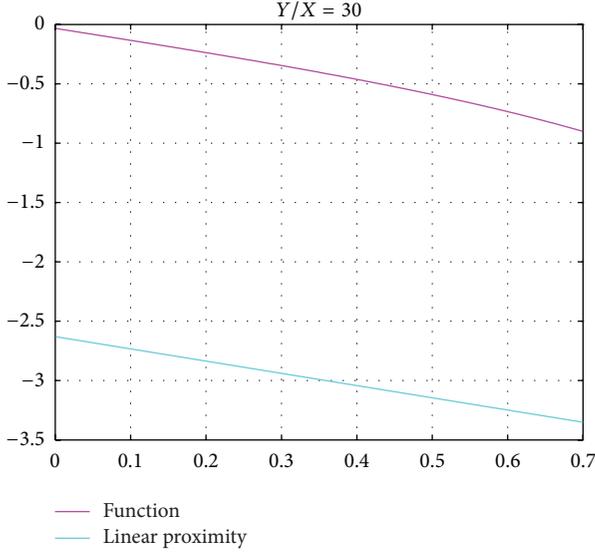


FIGURE 4: Pixel values vary with the yaw angle: our approach can successfully approximate the actual changes.

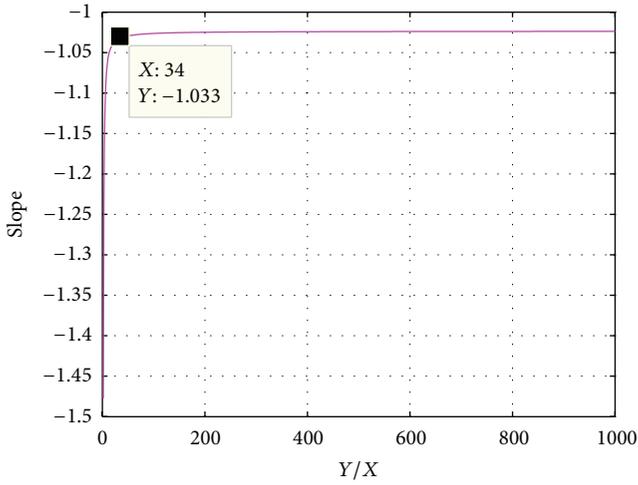


FIGURE 5: Slope converges to a stable value.

The motivation of the proposed method comes from the similar characteristic and phenomena between a camera and sonar measurement system and an OI algorithm. First, a camera and sonar system can be considered to be an OI system that produces an optimal output through several groups of observations. Second, the OI algorithm has the dimension-extensible and loose a priori characteristics that are attractive for camera and sonar measurement system.

The following are given:

- (i) A background field  $x_b$  available in two or three dimensions.
- (ii) A set of  $p$  observations  $x_{\text{obs}}$  available at irregular positions.

The optimal estimation is described by

$$x_a = x_b + W(x_{\text{obs}} - x_b). \quad (14)$$

TABLE 1: Initial state of camera before calibration.

Rotation angle	Pitch angle	Zoom
0	-15°	0

TABLE 2: Calibration results.

$f_x$	$f_y$	$U_0$	$V_0$
731.88	738.65	339.28	249.7

The errors are given by

$$\begin{aligned} \varepsilon^a &= x_a - x_t, \\ \varepsilon^b &= x_b - x_t, \\ \varepsilon^{\text{obs}} &= x_{\text{obs}} - x_t. \end{aligned} \quad (15)$$

The optimal weight is then as follows:

$$W = \frac{(\sigma^b)^2}{(\sigma^{\text{obs}})^2 + (\sigma^b)^2}, \quad (16)$$

where  $\sigma^a$  and  $\sigma^b$  represent the mean value of  $\varepsilon^a$  and  $\varepsilon^b$ . As data from camera and sonar are unrelated, it is assumed that  $E(\varepsilon^b \varepsilon^{\text{obs}}) = 0$ .

## 5. Evaluations and Analysis

In this section, we present the results of a set of physical experiments to demonstrate the performance of the proposed algorithm in Section 4. To validate the effectiveness of the proposed data rectification algorithm, we compared the results before and after a pose change with the truth. Moreover, we have conducted a set of experiments under different initial poses of a camera to testify the robustness of this method. Besides, comparative experiments have been designed to show the validity of the data assimilation approach.

Autonomous vehicles can be modeled as mobile robots and then we use the mobile robot Pioneer 3-DX (Figure 6) mounted with a camera for experiments. To prepare for experiments, the PTZ camera is firstly calibrated.

**5.1. Camera Calibration.** Grid size of the calibration board in experiments is 30 mm \* 30 mm. Picture resolution of VCC50I is fixed as 640 \* 480. To ensure error balance and calibration accuracy, a group of calibrated images containing four images from various poses are collected at a distance interval of 10 cm. Calibration distance ranges from 1500 mm to 4000 mm. Considering effects of pitch and rotation angles as well as the zoom value, the camera state during calibration is fixed as given in Table 1. Internal parameters are listed in Table 2, which are crucial to distance measurement.

**5.2. Performance Evaluation of the Data Rectification Algorithm.** Angle variations are acquired by a three-axis angle sensor, which act as the input of the rectification algorithm.



FIGURE 6: Pioneer 3-DX robot equipped with a Canon-VCC50I PTZ Camera.

TABLE 3: Different conditions set for measurement.

$\psi$	$\theta$	$\varphi$
2.13	7.18	0.05
4.89	4.84	2.81
5.68	4.05	3.59
2.93	6.42	2.77
4.07	5.01	1.10

The module (MPU6050) has advantages in low temperature dependency, high resolution, and low noise. Due to these advantages, this module is chosen as a tool to measure the Euler angles.

To validate the robustness of the algorithm, a target is set at different positions randomly. For each metric  $y/x$ , results under a set of camera poses are analyzed (yaw, pitch, and roll angles are random for each metric as set in Table 3). Initial readings of angle sensor are as follows:  $\theta$  (pitch):  $9.65^\circ$ ,  $\varphi$  (yaw):  $-0.27^\circ$ , and  $\psi$  (roll):  $-0.94^\circ$ . The target position tuples are set as  $\{(1500, 100), (2000, 100), (2500, 100), (3000, 100)\}$ .

Figure 7(a) shows measurements from the initial camera state. Compared with the truth, it shows bias along both horizontal and vertical directions.

The deviations of VO results caused by the camera motions are rectified independently. Based on the analysis of the range model in Section 3, the pitch angle is an independent variable of distance function. Therefore, we recalculate the pitch angle instead of inversion operations on pixel coordinates. Figures 7(b)–7(f) show results before and after rectification. In Figure 7(e), the distance error along the optical axis is almost as high as 50% using direct measurement. However, this value decreases to be only 6% using the proposed algorithm. We can also see that the least improvement in accuracy in this direction is 10% as shown in

Figure 7(c). Accordingly, much more remarkable effect can be seen from the results along the direction perpendicular to the optical axis. In the worst case as shown in Figure 7(f), the measured distance along  $X$ -axis is rectified from the measured  $-780$  mm to the final  $98$  mm. The percentage gains of measuring precision approach 878%. Even in a general situation, this percentage can be close to 35% as demonstrated in Figure 7(c). These figures also show that the range deviation becomes larger as the distance along the optical axis direction increases. This is mainly because the ratio of physical distance to pixel unit increases along the optical axis.

**5.3. Data Assimilation Evaluation.** For generality, the assimilation results under different metric size (i.e., manipulating  $y/x$ ) with a fixed camera pose are demonstrated. Data from sonar sensors are set as the background field value and those from camera are set as the observation field value. In Figure 8(a), measurement results at some positions are missing, which indicate that blind zones exist when sonar system only is adopted. Range data in Figure 8(b) are the results of sensor assimilation. It demonstrates the accuracy improvement from both  $X$ -axis and  $Y$ -axis compared with measurements solely from the vision system and sonar sensors. Assimilated results proved to be as much as 25 percent accurate along  $X$ -axis and 9 percent accurate along  $Y$ -axis compared to those acquired using a single type of sensor. This is mainly because new information is brought in to compensate the output from a single measuring system, that is, the wild FOV of camera and the high measurement accuracy of sonar sensors.

## 6. Conclusions

In this paper, we have proposed an analytical measuring rectification algorithm for monocular range estimation under camera shaking conditions. Specifically, we have established a pose-range model and then the algebraic relation between

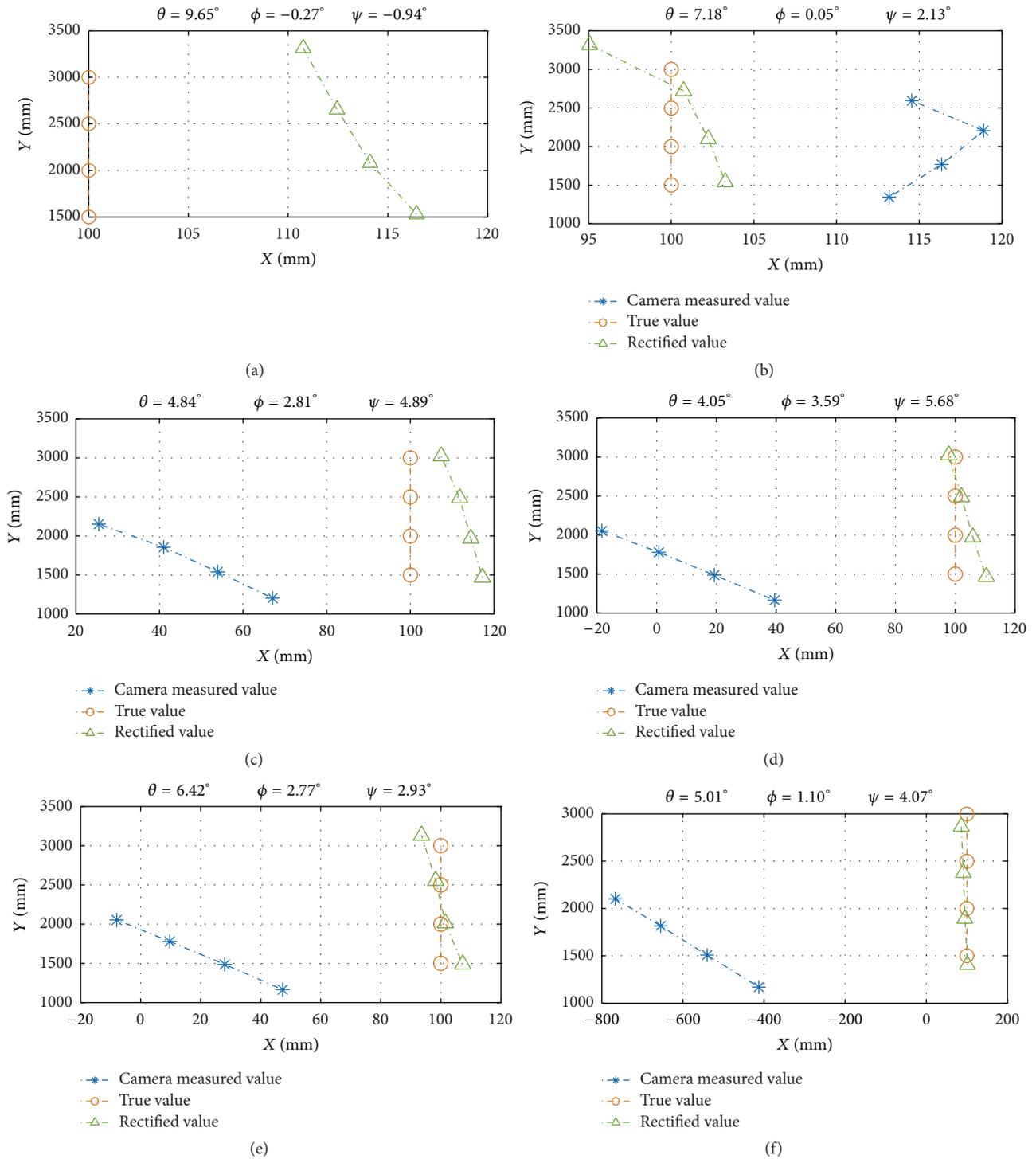


FIGURE 7: Comparisons between results after and before rectification: a set of camera poses are set to validate the robustness of designed algorithm.

distance increment and a camera's poses variation has been formulated. We have also designed a data assimilation system to provide reliable range information using different types of transducer systems. Physical experiments are conducted

to validate the effectiveness and robustness of the proposed algorithm. For future work we will try to implement our algorithms on the multiple robots formations as well as swarm coordination applications.

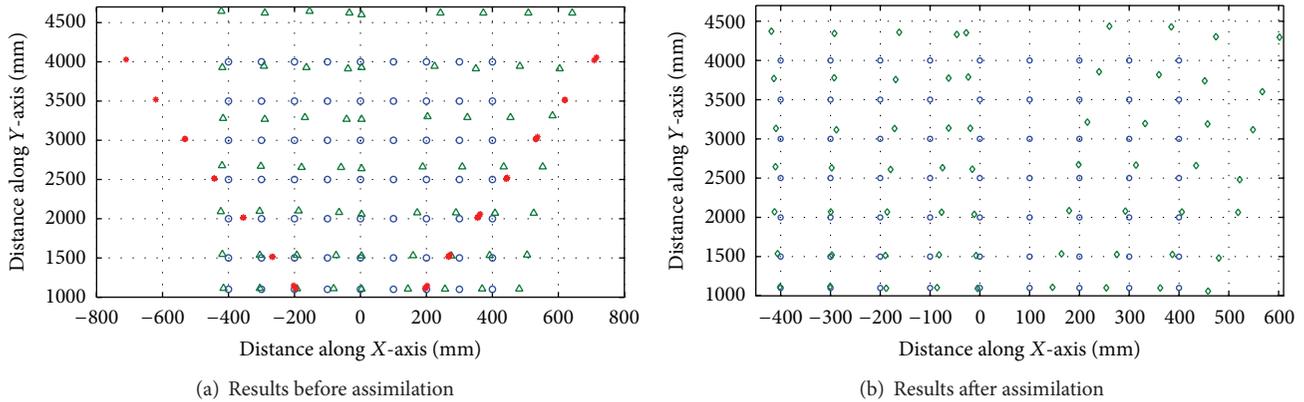


FIGURE 8: Comparisons between results before and after assimilation. (a) Original data from sensors of sonar and camera. (b) Results of assimilation using the Optimal Interpolation Algorithm. (Results of camera are represented as  $\Delta$  and those of sonar are described as  $*$ . The symbol  $\circ$  stands for truth values and  $\diamond$  are assimilated results.)

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

This research is financially supported by the Natural Science Foundation of China (Grant no. 61571334) and the Natural Science Foundation of China (Grant no. 2014AA09A512).

## References

- [1] M. Mikhalsky and J. Sitte, "Active motion vision for distance estimation in autonomous mobile robots," in *Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics*, vol. 2, pp. 1060–1065, December 2004.
- [2] N. Yamaguti, S. Oe, and K. Terada, "A method of distance measurement by using monocular camera," in *Proceedings of the 36th SICE Annual Conference (SICE '97)*, International Session Papers, pp. 1255–1260, IEEE, Tokushima, Japan, July 1997.
- [3] X. Liu, T. Qin, W. Chen, and X. Yin, "Real-time distance measurement using a modified camera," in *Proceedings of the IEEE Sensors Applications Symposium (SAS '10)*, pp. 54–58, IEEE, Limerick, Ireland, February 2010.
- [4] U. Mudenagudi and S. Chaudhuri, "Depth estimation using defocused stereo image pairs," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 1, pp. 483–488, IEEE, September 1999.
- [5] A. N. Rajagopalan, S. Chaudhuri, and U. Mudenagudi, "Depth estimation and image restoration using defocused stereo pairs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1521–1525, 2004.
- [6] M. N. A. Wahab, N. Sivadev, and K. Sundaraj, "Target distance estimation using monocular vision system for mobile robot," in *Proceedings of the 2nd IEEE International Conference on Open Systems (ICOS '11)*, pp. 11–15, IEEE, Langkawi, Malaysia, September 2011.
- [7] K. Umeda and T. Takahashi, "Subpixel stereo method: a new methodology of stereo vision," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, vol. 4, pp. 3215–3220, San Francisco, Calif, USA, April 2000.
- [8] T.-H. Wang, C.-C. Hsu, C.-C. Chen, C.-W. Huang, and Y.-C. Lu, "Three-dimensional measurement of a remote object with a single CCD camera," in *Proceedings of the 4th International Conference on Autonomous Robots and Agents (ICARA '09)*, pp. 187–192, IEEE, Wellington, New Zealand, February 2009.
- [9] K. Nakamura, K. Ishigaki, T. Ogata, and S. Muramatsu, "Real-time monocular ranging by Bayesian triangulation," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '13)*, pp. 1368–1373, IEEE, Gold Coast, Australia, June 2013.
- [10] A. Tsalatsanis, K. Valavanis, and N. Tsourveloudis, "Mobile robot navigation using sonar and range measurements from uncalibrated cameras," *Journal of Intelligent and Robotic Systems*, vol. 48, no. 2, pp. 253–284, 2007.
- [11] V. Tucakov, M. Sahota, D. Murray et al., "A stereoscopic visually guided mobile robot," in *Proceedings of the Hawaii International Conference on Systems Sciences*, Maui, Hawaii, USA, January 1997.
- [12] B. Fidan, V. Gazi, S. Zhai, N. Cen, and E. Karatas, "Single-view distance-estimation-based formation control of robotic swarms," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, pp. 5781–5791, 2013.
- [13] X. Li and L. Wang, "A monocular distance estimation method used in video sequence," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA '12)*, pp. 390–394, Shenyang, China, June 2012.
- [14] J. Yu, J. Song, X. Sun, and T. Zhang, "Design and experiment of distance measuring system with single camera for picking robot," in *Proceedings of the IEEE World Automation Congress (WAC '10)*, pp. 445–448, Kobe, Japan, September 2010.
- [15] L. Guo, Y. Xu, K. Li et al., "Study on real-time distance detection based on monocular vision technique," *Journal of Image and Graphics*, vol. 11, no. 1, pp. 74–81, 2006.
- [16] S. Derrouch, K. Izumida, and K. Shiiya, "A combination of monocular CCD camera and inertial-sensor for range estimation," in *Proceedings of the 28th Annual Conference of the IEEE Industrial Electronics Society (IECON '02)*, pp. 2191–2196, IEEE, November 2002.

- [17] D. Wang, W. Chen, Z. Zhao, and T. C. Ng, "Inter-vehicle separation measurement using monocular vision," in *Proceedings of the 7th IEEE Conference on Industrial Electronics and Applications (ICIEA '12)*, pp. 258–263, Singapore, July 2012.
- [18] Y. X. Han, Z. S. Zhang, and M. Dai, "Monocular vision system for distance measurement based on feature points," *Optics and Precision Engineering*, vol. 19, no. 5, pp. 1110–1117, 2011.
- [19] E. Malis and P. Rives, "Robustness of image-based visual servoing with respect to depth distribution errors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '03)*, vol. 1, pp. 1056–1061, IEEE, Taipei, Taiwan, September 2003.
- [20] R. F. Stengel, *Optimal Control and Estimation*, Dover, New York, NY, USA, 1994.
- [21] J. Kim, Y. Kim, and S. Kim, "An accurate localization for mobile robot using extended Kalman filter and sensor fusion," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '08)*, pp. 2928–2933, IEEE World Congress on Computational Intelligence, Hong Kong, June 2008.

## Research Article

# Using Omnidirectional Vision to Create a Model of the Environment: A Comparative Evaluation of Global-Appearance Descriptors

L. Payá, O. Reinoso, Y. Berenguer, and D. Úbeda

*Department of Systems Engineering and Automation, Miguel Hernández University, Avenida de la Universidad s/n, Elche, 03202 Alicante, Spain*

Correspondence should be addressed to L. Payá; [lpaya@umh.es](mailto:lpaya@umh.es)

Received 23 October 2015; Accepted 11 February 2016

Academic Editor: Yassine Ruichek

Copyright © 2016 L. Payá 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.

Nowadays, the design of fully autonomous mobile robots is a key discipline. Building a robust model of the unknown environment is an important ability the robot must develop. Using this model, this robot must be able to estimate its current position and to navigate to the target points. The use of omnidirectional vision sensors is usual to solve these tasks. When using this source of information, the robot must extract relevant information from the scenes both to build the model and to estimate its position. The possible frameworks include the classical approach of extracting and describing local features or working with the global appearance of the scenes, which has emerged as a conceptually simple and robust solution. While feature-based techniques have been extensively studied in the literature, appearance-based ones require a full comparative evaluation to reveal the performance of the existing methods and to tune correctly their parameters. This work carries out a comparative evaluation of four global-appearance techniques in map building tasks, using omnidirectional visual information as the only source of data from the environment.

## 1. Introduction

During the last years, the presence of mobile robots in both industrial and household environments has increased substantially since they are able to solve many different tasks. The expansion of robots into such environments and applications has been eased thanks to the development of their abilities in perception, computation, autonomy, and adaptability to different circumstances. As far as perception is concerned, the robots must be equipped with sensors that allow them to extract the necessary information from the environment to be able to carry out autonomously their tasks. Vision sensors have gained popularity because they present some interesting advantages such as providing a big quantity of information with a relatively low cost and low power consumption (comparing to other sensors, such as laser rangefinders) and stable data both outdoors and indoors (unlike GPS, whose signal is prone to degradation indoors). They also permit carrying out additional high level tasks, such as people detection and recognition. Among vision sensors,

catadioptric systems have extended in recent years as they are able to capture images with a field of view of 360 deg. around the robot [1]. In our approach, the mobile robot is equipped with a catadioptric system on it, which captures images from the environment. Using this information, the objective is building a robust model of the environment. In general, these models can be represented as a metric, a topological, or a hybrid map [2]. First, metric maps define the position of some relevant features of the environment with respect to a coordinate system and permit robot localization with geometric accuracy (except for a relative error) [3]. Second, topological maps often represent the environment as a graph where nodes are distinctive localizations (e.g., rooms) of the environment and links are the connectivity relationships between localizations. Usually, such maps do not permit fine localization but they are enough to estimate the position of the robot and to navigate to the desired localizations [4]. At last, hybrid maps are hierarchical models where information is arranged in multiple levels. Usually, there is a high level of topological information that allows an

approximate localization (in an area of the environment) and several low levels of metric information that permit refining the localization (in the previously detected area) [5].

In all cases, to build a functional map, it is necessary to extract relevant information from the scenes. Traditionally, researchers have focused on methods that extract some outstanding landmarks or interest points from the scenes and describe them with any robust description method. These methods have become popular in map building and mobile robots localization. For example, Angeli et al. [6] make use of SIFT features [7] to solve the mapping and global localization problems simultaneously (SLAM) and Valgren and Lilienthal [8] and Murillo et al. [9] make use of SURF features [10] to solve the localization problem in a previously created model. Using feature-based approaches in combination with probabilistic techniques, it is possible to build metric maps [3]. However, these methods present some drawbacks; for example, it is necessary that the environment be rich in prominent details (otherwise, artificial landmarks can be inserted in the environment, but this is not always possible); also, the detection of such points is sometimes not robust against changes in the environment and their description is not always invariant to changes in robot position and orientation. Besides, camera calibration is crucial in order to incorporate new measures in the model correctly. This way, small deviations in either the intrinsic or the extrinsic parameters add some error to the measures. At last, extracting, describing, and comparing landmarks are computationally complex processes that often make building the model in real time unfeasible, as the robot explores the environment.

In contrast, global-appearance techniques have gained relevance in more recent works [11–13]. These techniques are useful when the robot moves within unstructured environments where extracting and describing robust points is difficult. These approaches lead to conceptually simpler algorithms since each scene is described by means of a unique descriptor. Map creation and localization can be achieved just storing and comparing pairwise these descriptors. As a drawback, extracting metric relationships from this information is difficult; thus, this family of techniques is usually employed to build topological maps (unless the visual information is combined with other sensory data, such as odometry). Despite their simplicity, several difficulties must be faced when using these techniques. Since no local information is extracted from the scenes, it is necessary to use any compression and description method that make the process computationally feasible. These descriptors do not present invariance to changes neither in the robot orientation nor in the lighting conditions or other changes in the environment (position of objects, doors, etc.). They will also suffer problems in environments where *visual aliasing* is present, which is a common phenomenon in indoor environments with repetitive visual structures.

Many algorithms can be found in the literature working both with local features and with global appearance of images. All these algorithms imply many parameters that have to be correctly tuned so that the mapping and localization processes are correct. Feature-based approaches have reached a relative maturity and some comparative evaluations have

been carried out, such as [14]. These evaluations are useful to choose the most suitable extractor and descriptor to a specific application. However, global-appearance-based approaches are still a field that is worth deeper exploration. We have not found any work that makes a comparative evaluation of the performance of such descriptors in mapping tasks. This is the main objective we propose in this paper. We have selected four accepted global-appearance description methods, adapted them to be used with omnidirectional visual information, and studied their properties. Then, we have developed the necessary algorithms to create a model of the environment, tested their performance, and studied the influence of the most relevant parameters.

The remainder of the paper is structured as follows. Section 2 presents briefly the description approaches that will be evaluated along the paper. After that, Section 3 describes the kind of models of the environment we will build to test the performance of the approaches. Then, Section 4 details the set of experiments designed and the results obtained. To finish, a final discussion is carried out in Section 5.

## 2. Global-Appearance Descriptors: State of the Art

This section outlines some methods to describe the global appearance of images. Four families of methods are proposed to be analysed: methods based on the discrete Fourier transform (Section 2.1), on principal components analysis (Section 2.2), on orientation gradients (Section 2.3), and on the essence of the scenes (Section 2.4). These are the description methods whose performance will be evaluated along the paper.

### 2.1. Methods Based on the Discrete Fourier Transform (DFT).

The Discrete Fourier Transform (DFT) is a classical method to describe scenes that presents some interesting features. When the two-dimensional DFT of a scene  $im(x, y)$  is calculated, the result is a complex function  $IM(u, v)$  in the frequency domain ( $u$  and  $v$  are the frequency variables) that can be decomposed into magnitude and argument matrices. The first matrix (also known as amplitude spectrum) represents the distribution of spatial frequencies within the image (i.e., it contains information on the overall structure of the image: edges orientation, smoothness, width, etc.). On the other hand, the argument matrix contains information about the local properties of the scene (shape and position of the objects). Taking these facts into account, the amplitude spectrum can be used as a global descriptor of the scene, since it contains information about the dominant structural patterns and it is invariant to the distribution of the objects. This information has proved to be relevant to solve simple classification tasks [15]. However, this kind of descriptors has no information about the spatial relationships between the structures in the image. To have a complete description of the scene, such information must be included.

Considering it, we have opted for a formulation of the DFT which contains complete information. This formulation is the Fourier Signature (FS), described first in [12]. It is defined as the matrix composed of the 1D DFT of each row

in the original image. When applied to panoramic scenes, it offers rotational invariance. When we calculate the FS of a panoramic image  $\text{im}(x, y) \in \mathbb{R}^{N_x \times N_y}$ , we arrive to a new matrix  $\text{IM}(u, y) \in \mathbb{C}^{N_x \times N_y}$ , where the main information is concentrated in the low frequency components of each row (so we can retain only the  $k$  first columns, having a compression effect). This new matrix with  $N_x$  rows and  $k$  columns can be decomposed in a magnitude matrix  $A_j(u, y) = |d_j(u, y)|$  with  $N_x$  rows and  $k_1$  columns and an argument matrix  $\Phi_j(u, y)$ , with  $N_x$  rows and  $k_2$  columns.

Based on the shift property of the DFT, when two panoramic images have been captured from the same position but have the robot different orientations, both images have the same magnitude matrix and the arguments matrices permit obtaining the relative robot orientation. This property allows us to use the magnitude matrix to estimate the position of the robot (as it presents rotational invariance) and, then, the arguments matrix to estimate the relative orientation of the robot.

### 2.2. Methods Based on Principal Components Analysis (PCA).

Panoramic images are data that fall in a space with a very high number of dimensions. However, the image pixels tend to be very correlated data, since they have been captured from a 3DOF process (robot pose on the ground plane). Taking this fact into account, a natural way to compress the information is principal components analysis (PCA), as shown in [16]. This kind of descriptors has evolved from the original formulation to adapt them to be used in mapping and localization tasks. The works of Leonardis and Bischof [17] show some examples of how this analysis can be used to mobile robots localization in a robust way.

When we have a set of  $n$  images  $\text{im}_j(x, y) \in \mathbb{R}^{N_x \times N_y}$ ,  $j = 1, \dots, n$ , each image can be considered a point in a space with  $N_x \cdot N_y$  dimensions,  $\vec{x}_j(i) \in \mathbb{R}^{N_x \cdot N_y \times 1}$ ,  $j = 1, \dots, n$ , ( $n \ll N_x \cdot N_y$ ). Using the classical formulation of PCA, it is possible to transform each point  $\vec{x}_j(i)$ , in a new data point, namely, *image projection*  $\vec{p}_j(i) \in \mathbb{R}^{k_3 \times 1}$   $j = 1, \dots, n$ , where  $k_3$  is the number of PCA features that contain the most relevant information,  $k_3 \leq n$ . Turk and Pentland [18] show how the necessary transformation matrix  $\mathbf{V}$  can be obtained in an efficient way. They make use of the SVD of the data matrix covariance, retaining only the eigenvectors with higher eigenvalues. If the number of eigenvectors is equal to  $n$ , then there is no loss of information during the compression process [16]. Thus, after applying PCA techniques, images can be handled efficiently, with a low computational cost. However, depending on the images' size, the process to obtain  $\mathbf{V}$  may be substantially slow.

The use of PCA in mapping and localization tasks is limited since the image projections depend on the robot orientation. Independently, on using omnidirectional scenes, the images projections contain only information of the position and orientation the robot had when capturing the images. This is the reason why Jogan and Leonardis developed the concept of *eigenspace of spinning images* [19]. This model uses specific properties of panoramic images to obtain, in an efficient way, an optimal subspace that takes into account the different orientations a robot may have

when capturing each image. The method takes profit of the symmetry properties the data matrix presents when we add the rotations information. This method has the advantage of permitting the estimation of the robot orientation, but the computational cost to obtain the transformation matrix  $\mathbf{V}$  is extremely high. By this reason, it has been only used with small environments, with a limited number of images.

### 2.3. Methods Based on the Histogram of Oriented Gradients (HOG).

HOG is a description method used traditionally in object detection. This technique considers the gradient orientation in localized parts of a scene. The method outstands by its simplicity, good computational cost, and relatively good results in object recognition tasks. It was initially described by Dalal and Triggs [20], who used it in people detection tasks. Later on, some researchers developed an improved version of the algorithm both in detection accuracy and in computational cost [21].

However, the experience with HOG descriptors in the mobile robotics field is limited to simple and small environments. Few previous works have made use of HOG in robot mapping and localization. Hofmeister et al. [22] use this descriptor in small robots localization tasks, with low resolution images and small environments not prone to *visual aliasing*. Under these limited conditions, the algorithm works well.

HOG is not defined as a global-appearance descriptor because the basic implementation consists in dividing the scene in a set of cells and obtaining a histogram of gradient orientation using the pixels information in each cell. The combination of all these histograms is the image descriptor. We have redefined the algorithm to obtain a unique descriptor per image that contains information of the global appearance of this image. The version of HOG we consider is described in [23], where a global version of HOG is used to carry out mapping and Monte-Carlo localization in large environments. Anyway, it is necessary to make an evaluation of the performance of this algorithm and systematize it in map creation tasks.

### 2.4. Methods Based on Gist and Prominence.

The *gist* concept was first introduced by Oliva and Torralba [24], with the idea of creating a low-dimension scene descriptor, and avoiding segmentation and processing of points, objects, or individual regions. They inspired by some works that suggested that humans recognize scenes by codifying the global configuration and just ignoring most of the details and individual objects [25].

More recently, some works make use of the *prominence* concept together with *gist*. It refers to regions of pixels that stand out with respect to the neighbor regions, in contrast to *gist*, which implies the accumulation of statistical data from the whole image. Siagian and Ilti [26] try to establish a synergy between the two concepts and they design a unique descriptor that takes both into account. This descriptor is built using the intensity, orientation and color information.

The experience with this kind of descriptors in mobile robots applications is limited. For example, Chang et al. [27] present a localization and navigation system based on *gist* and *prominence* and Murillo et al. [28] make use of *gist*

descriptors in a localization problem. However, they obtain these descriptors using specific regions in a set of panoramic images.

Like HOG, *gist* is not primarily defined as a global-appearance descriptor and we have redefined the algorithm to obtain a unique descriptor per image. The version of *gist* we consider in this evaluation is described in [23] and is built from orientation information, analysed in some resolution levels.

### 3. Creating a Visual Topological Map of the Environment

In this section we focus on the map creation problem. The robot, which is equipped with a catadioptric vision system on its top, explores the environment to map to cover it completely. During this process, the robot captures a set of omnidirectional scenes from several positions. Only this visual information will be used to build the map (neither odometry nor laser or other sensory data will be used). This way, the final model will be a topological map since it contains some localizations (represented as panoramic scenes) and connectivity relations, but no metric data. In Section 3.1, we describe how the nodes of the map are represented with each description method and, in Section 3.2, the process to add connections between the nodes is outlined.

*3.1. Using Global-Appearance Descriptors to Create a Model of the Environment.* Let us suppose that the mobile robot has gone across the environment to map (either in a teleoperated way or autonomously, following any exploration algorithm) and has captured a set of omnidirectional images  $I = \{im_1, im_2, \dots, im_n\}$ , where  $im_j \in \mathbb{R}^{N_x \times N_y}$ .

From this set of images, a set of descriptors, one per original scene, is calculated. As a result, the nodes of the map will be a set of descriptors  $D = \{d_1, d_2, \dots, d_n\}$  where, in general,  $d_j \in \mathbb{C}^{M_x \times M_y}$ . With the objective that these nodes are functional, it is necessary that  $d_i$  contains information that permits estimating the position of the robot when capturing  $im_i$  (taking into account that the robot may have any orientation in this position). In the next subsections, we detail the kind of information each  $d_i$  should contain when using each description method.

*3.1.1. DFT Descriptor.* Each node  $d_i$  contains two matrices: the magnitudes one  $A_i(u, y) \in \mathbb{R}^{N_x \times k_1}$  and the arguments matrix  $\Phi_i(u, y) \in \mathbb{R}^{N_x \times k_2}$ .  $k_1$  is the number of columns we retain in the localization descriptor and  $k_2$  is the number of columns retained in the orientation descriptor. The higher  $k_1$  and  $k_2$ , the more information the descriptor contains. However, we must take into account that the main information is concentrated in the low frequency columns, and if noise is present on the image, it will affect high frequency components mostly; thus, removing these components may imply an additional benefit. The effect of both parameters in a mapping process will be evaluated.

*3.1.2. PCA Descriptor.* The PCA descriptor we use is proposed in the works of Jogan and Leonardi [19]. This model uses

the specific properties of panoramic images to create a set of  $N_R$  spinning images from each of the  $n$  original panoramic images, so we get  $N_R$  data vectors per original image. To obtain the transformation matrix  $\mathbf{V}$ , the similarities among the rotated versions of each image are taken into account. This permits decomposing the original problem (which is computationally very heavy) in a set of lower order problems.

As a result of the process, the map will be composed of (a) a set of descriptors  $\vec{p}_j(i) \in \mathbb{R}^{k_3 \times 1}$   $j = 1, \dots, n$ , which are the projections of the original panoramic images and contain information on the robot position, (b) a set of phase vectors,  $\vec{\phi}_j \in \mathbb{R}^{k_3 \times 1}$ , one per image, which contain information of the robot orientation, and (c) a unique transformation matrix  $\mathbf{V} \in \mathbb{C}^{k_3 \times N_x \cdot N_y}$ .  $k_3$  is the number of eigenvector chosen. The higher the  $k_3$ , the more the information that the map contains. If  $k_3 = n$ , there is no loss of information.

*3.1.3. HOG Descriptor.* Each image will be described through two HOG descriptors. The first one,  $\vec{h}_1$ , is the position descriptor and is invariant against rotations of the robot. To obtain it, the panoramic image is divided into horizontal cells, whose width is equal to  $N_y$  (number of columns in the image) and whose height can be configured freely. The size of  $\vec{h}_1$  is  $1 \times k_4 \cdot b$ , where  $k_4$  is the number of horizontal cells and  $b$  is the number of bins in each orientation histogram. The second one,  $\vec{h}_2$ , is the orientation descriptor. To obtain it, the panoramic image is divided into vertical cells whose height is equal to  $N_x$ . Some overlap between these cells may exist. If the width of the cells is  $l_1$  and the distance between consecutive cells  $d_1$ , then the number of vertical cells is  $k_5 = N_y/d_1$ . The size of the orientation descriptor  $\vec{h}_2$  is then  $1 \times k_5 \cdot b$ . In the experiments, the influence of  $k_4$ ,  $k_5$ , and  $b$  will be evaluated.

Figure 1 shows, from a panoramic image whose gradient has been calculated, the process to obtain both descriptors: (a)  $\vec{h}_1$  and (b)  $\vec{h}_2$ .

*3.1.4. Gist and Prominence Descriptor.* The information of the orientation of the edges in the image is used to build the descriptor. First, two versions of each image are considered: the original one and a new version after applying a Gaussian low-pass filter and subsampling to a new size  $0.5N_x \times 0.5N_y$ . Second, both images are filtered with a bank of  $m$  Gabor filters with orientations evenly distributed between 0 and 180 deg. Third, to reduce the amount of information, the pixels in each resulting image are grouped into blocks. The block division is carried out in a similar fashion as in HOG: a position descriptor  $\vec{g}_1$  is obtained by defining  $k_6$  horizontal blocks and an orientation descriptor  $\vec{g}_2$  is calculated with  $k_7$  vertical blocks (with overlapping). In the experiments, the influence of  $k_6$ ,  $k_7$ , and  $m$  will be evaluated. Figure 2 shows, from a panoramic image, the process to obtain  $\vec{g}_1$ .

To sum up, Table 1 shows the parameters to be tuned in each description method included in the evaluation. On the other hand, Table 2 gives details of the contents of the map when we consider each description method.

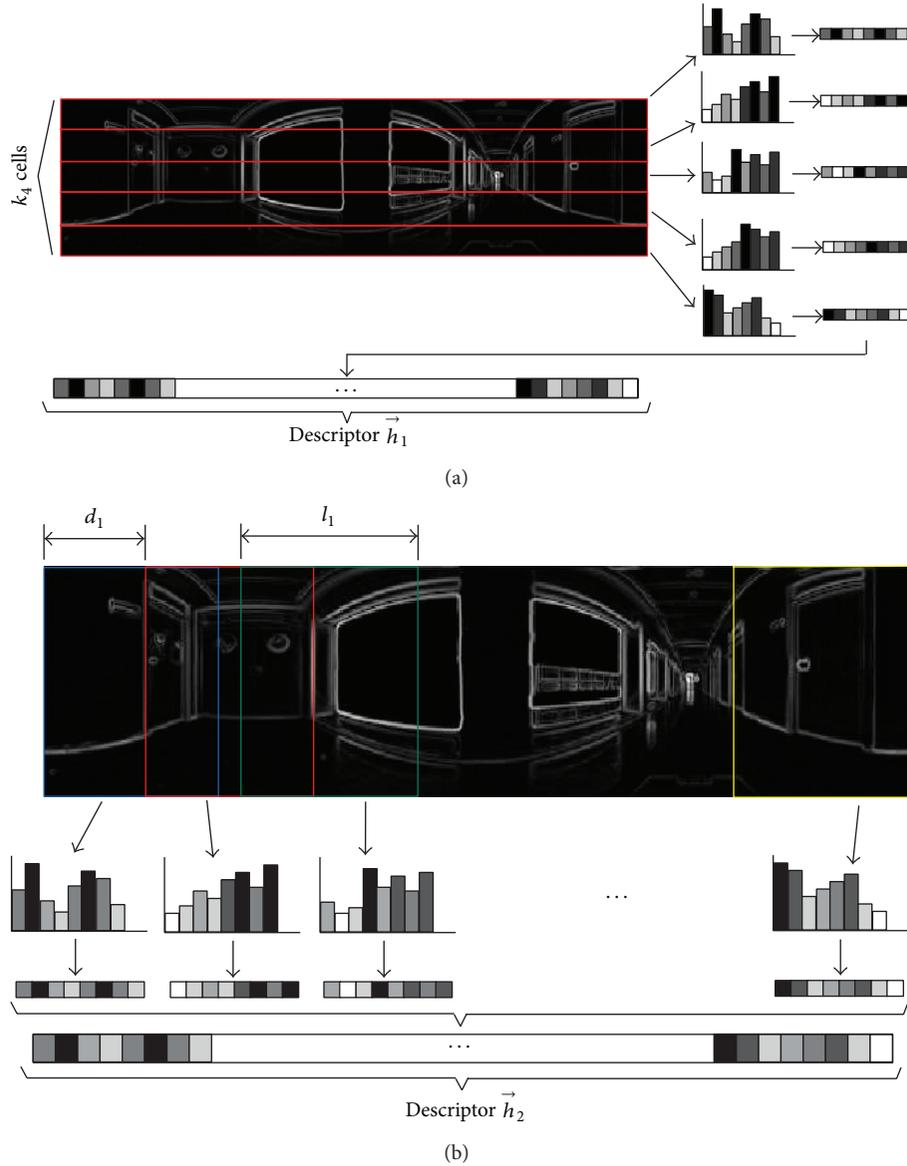


FIGURE 1: Process to obtain (a) the HOG position descriptor  $\vec{h}_1$  and (b) the HOG orientation descriptor  $\vec{h}_2$ .

**3.2. Adding Topological Relations.** Our starting point is a set of images captured from unknown positions. The objective of this section consists in designing an algorithm that allows us to establish adjacency relations among them, with the goal of creating a topological map. Apart from this, we expect the distribution of the nodes in this map to be similar to the distribution of the points where the images were captured. It goes beyond the classical concept of topological map since besides adjacency it also introduces the concepts of closeness and farness. Thanks to this kind of maps, the robot will be able to plan its trajectory more accurately.

To create such a map, a method based on a mechanical system of forces is used. This kind of methods has been used often to simulate the movement of flexible bodies, as in [29],

where the body is discretized into a set of particles, and the interaction among them is modelled with a set of springs. Our framework also includes a set of dampers in parallel with the springs, since the dampers can help to achieve an overdamped behaviour that facilitates reaching the steady state.

The idea we develop consists in considering each image a particle which is linked to the rest of images (particles) through a pair spring-damper, where the natural length of each spring is equal to the distance between the descriptors of the two images linked by this spring. The particles start their evolution from random positions. If we let the forces produced by springs and dampers move freely in the system until it tends to a minimum energy position, we expect the distribution of particles to be similar to the distribution

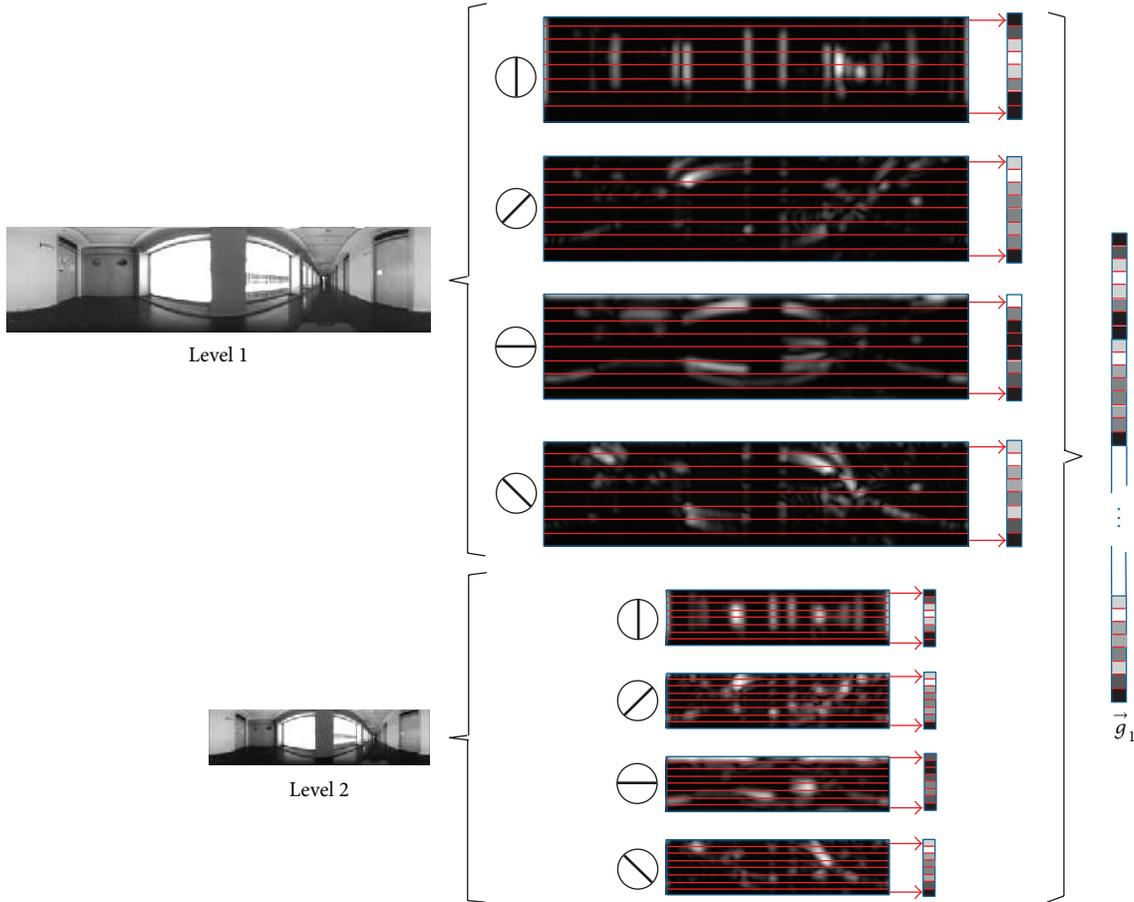


FIGURE 2: Process to build the gist position descriptor  $\vec{g}_1$  ( $m = 4$  orientations,  $k_6 = 8$  blocks).

of capture points. The algorithm we use is inspired by the algorithm presented by Menegatti et al. [12], who used it in small environments.

**3.2.1. Mass-Spring-Damper Method.** Each image is considered a particle  $P_j$ ,  $j = 1, \dots, n$ , with mass  $m_j$ , where  $n$  is the number of images to include in the map. No information about the coordinates of the capture points is available.

Each pair of particles  $P_k$  and  $P_l$  is linked with a spring  $S_{kl}$  with elastic constant  $k_{kl}$  and a damper with damping constant  $\kappa_{kl}$ . The natural length of each spring  $l_{kl}^0$  is equal to the distance between the descriptors of the images associated with the particles  $P_k$  and  $P_l$ .

The initial positions of the particles are randomly initialised. After that, the system is allowed to evolve freely until it reaches a steady state. At this state, the distribution of the particles is expected to be similar to the distribution of capture points (except for a scale factor and a rotation). This way, the result is a scaled model of the real distribution. We consider the value of the elastic constants to be proportional to the distance between images and, from a threshold distance, the images are not linked by any spring.

Under these circumstances, the spring and damper linking each pair of particles  $P_i$  and  $P_j$  make on these particles the force:

$$\vec{f}_{ij} = - \left[ -k_{ij} \cdot (|\vec{p}_i - \vec{p}_j| - l_{ij}^0) + \kappa_{ij} \cdot \left( \frac{(\vec{v}_i - \vec{v}_j) \cdot (\vec{p}_i - \vec{p}_j)}{|\vec{p}_i - \vec{p}_j|} \right) \right] \cdot \frac{(\vec{p}_i - \vec{p}_j)}{|\vec{p}_i - \vec{p}_j|}, \quad (1)$$

$$\vec{f}_{ji} = -\vec{f}_{ij},$$

where  $\vec{p}_i, \vec{v}_i$  are the position and speed of the  $i$ th particle, respectively. Then, the resulting force on each particle is obtained:

$$\vec{F}_i = \sum_{j=\{1, \dots, n\}, j \neq i} \vec{f}_{ij}. \quad (2)$$

TABLE 1: Parameters to be tuned in each description method.

Descriptor	Parameters
FS	$k_1 \Rightarrow$ components per row, localization descriptor $\mathbf{A}_j$
	$k_2 \Rightarrow$ componentes per row, orientation descriptor $\Phi_j$
r-PCA	$k_3 \Rightarrow$ number of eigenvectors
	$N_R \Rightarrow$ number of rotations per panoramic image
HOG	$b \Rightarrow$ number of bins per histogram
	$k_4 \Rightarrow$ number of horizontal cells, localization descriptor $\vec{h}_1$
	$l_1 \Rightarrow$ width of vertical cells, orientation descriptor $\vec{h}_2$
	$d_1 \Rightarrow$ distance between vertical cells, orientation descriptor $\vec{h}_2$
	$k_5 = N_y/d_1 \Rightarrow$ number vertical cells, orientation descriptor $\vec{h}_2$
Gist	$m \Rightarrow$ number of orientations (Gabor filters), localization descriptor $\vec{g}_1$
	$k_6 \Rightarrow$ number of horizontal blocks, localization descriptor $\vec{g}_1$
	$l_2 \Rightarrow$ width of vertical blocks, orientation descriptor $\vec{g}_2$
	$d_2 \Rightarrow$ distance between vertical blocks, orientation descriptor $\vec{g}_2$
	$k_7 = N_y/d_2 \Rightarrow$ number of vertical blocks, orientation descriptor $\vec{g}_2$

TABLE 2: Contents of the map, relative to localization and orientation estimation, per image included in the model  $\text{im}_j$ ,  $j = 1, \dots, n$ .

Descriptor	Localization	Orientation
FS	$\mathbf{A}_j \in \mathbb{R}^{N_x \times k_1}$	$\Phi_j \in \mathbb{R}^{N_x \times k_2}$
r-PCA	$\mathbf{V} \in \mathbb{C}^{k_3 \times N_x \times N_y}$ $\vec{p}_j \in \mathbb{C}^{k_3 \times 1}$	$\vec{\phi}_j \in \mathbb{R}^{k_3 \times 1}$
HOG	$\vec{h}_{1j} \in \mathbb{R}^{k_4 \cdot b \times 1}$	$\vec{h}_{2j} \in \mathbb{R}^{k_5 \cdot b \times 1}$
Gist	$\vec{g}_{1j} \in \mathbb{R}^{2 \cdot k_6 \cdot m \times 1}$	$\vec{g}_{2j} \in \mathbb{R}^{k_7 \cdot m \times 1}$

From this resulting force, the acceleration of the particle is obtained from the 2nd Newton's law:

$$\vec{a}_i(t) = \frac{\vec{F}_i}{m_i}, \quad (3)$$

where  $\vec{a}_i(t)$  is the  $i$ th particle acceleration at time instant  $t$ ,  $\vec{F}_i$  is the resulting force on particle  $i$ , and  $m_i$  is the  $i$ th particle mass. From this acceleration, the speed and position of particle  $i$  once it passed a period of time  $\Delta t$  can be calculated:

$$\begin{aligned} \vec{v}_i(t + \Delta t) &= \vec{v}_i(t) + \vec{a}_i(t) \cdot \Delta t, \\ \vec{p}_i(t + \Delta t) &= \vec{p}_i(t) + \vec{v}_i(t) \cdot \Delta t. \end{aligned} \quad (4)$$

This method, known as Euler integration, may not be stable if the step time is not low enough, which would increase the computational cost of the process. This is the reason why the Verlet integration is sometimes suggested. In this

integration method, the position and speed are updated at each iteration with the following expressions.

At the time instant  $t = \Delta t$ ,

$$\begin{aligned} \vec{p}_i(\Delta t) &= \vec{p}_i(0) + \vec{v}_i(0) \cdot \Delta t + 0.5 \cdot \vec{a}_i(0) \cdot \Delta t^2, \\ \vec{v}_i(\Delta t) &= \vec{v}_i(0) + \vec{a}_i(0) \cdot \Delta t. \end{aligned} \quad (5)$$

From this time instant,

$$\begin{aligned} \vec{p}_i(t + \Delta t) &= 2 \cdot \vec{p}_i(t) - \vec{p}_i(t - \Delta t) + \vec{a}_i(t) \cdot \Delta t^2, \\ \vec{v}_i(t + \Delta t) &= \frac{\vec{p}_i(t + \Delta t) - \vec{p}_i(t)}{2 \cdot \Delta t}. \end{aligned} \quad (6)$$

## 4. Experiments

In this section, we compare the performance of the four description methods. First, we describe the sets of images we have used to carry out the experiments. Then, the evaluation is carried out from several points of view to fully uncover the goodness of each method in mapping tasks. We analyse the computational cost of the mapping process, the relationship between the image distance and the geometric distance, and the performance in topological map building.

*4.1. Sets of Images.* To carry out the experiments, we make use of two sets of images, captured with two different catadioptric systems. First, set 1 has been captured by us in a building of Miguel Hernández University (Spain). The images were captured along 6 different rooms in an office-like environment. Figure 3(a) shows a bird's eye view of this environment. The database is composed of 873 panoramic  $64 \times 256$ -color images which have been captured on a dense  $40 \times 40$  cm grid of points (red points in Figure 3(a)). Set 1 [30] is a challenging database due to the tendency to visual aliasing that presents the environment. There are many zones which, despite being geometrically far, present a similar visual appearance. Also, the images were captured in different times of day (changing lighting conditions) and the positions of some objects in the scenes are modified (e.g., changes in the state of doors). All the images were captured with an *Imaging Source DFK 21BF04* camera mounted on a *Pioneer 3-AT* robotic platform. The camera takes pictures of a hyperbolic mirror (*Eizoh Wide 70*) which is mounted on it with its axis aligned with the camera optic axis. The resulting omnidirectional images are transformed with a cylindrical projection to obtain their panoramic versions. The *P3-AT* robot has 4 drive wheels. Its maximum linear speed is equal to 0.7 m/s, its maximum turning speed is equal to 140 deg/s, and the minimum turning radius is null. The robot can move freely on the floor so the image capture process has 3 degrees of freedom: position on the ground floor with respect to a world coordinate system  $(x, y)$  and orientation with respect to the  $z$ -axis  $(\theta)$ . Figure 3(b) shows the robot, the catadioptric system, and a sample image (omnidirectional and panoramic formats).

The second set of images has been captured by a third party [31]. It is composed of a set of panoramic grayscale images, captured in several rooms of a university and a

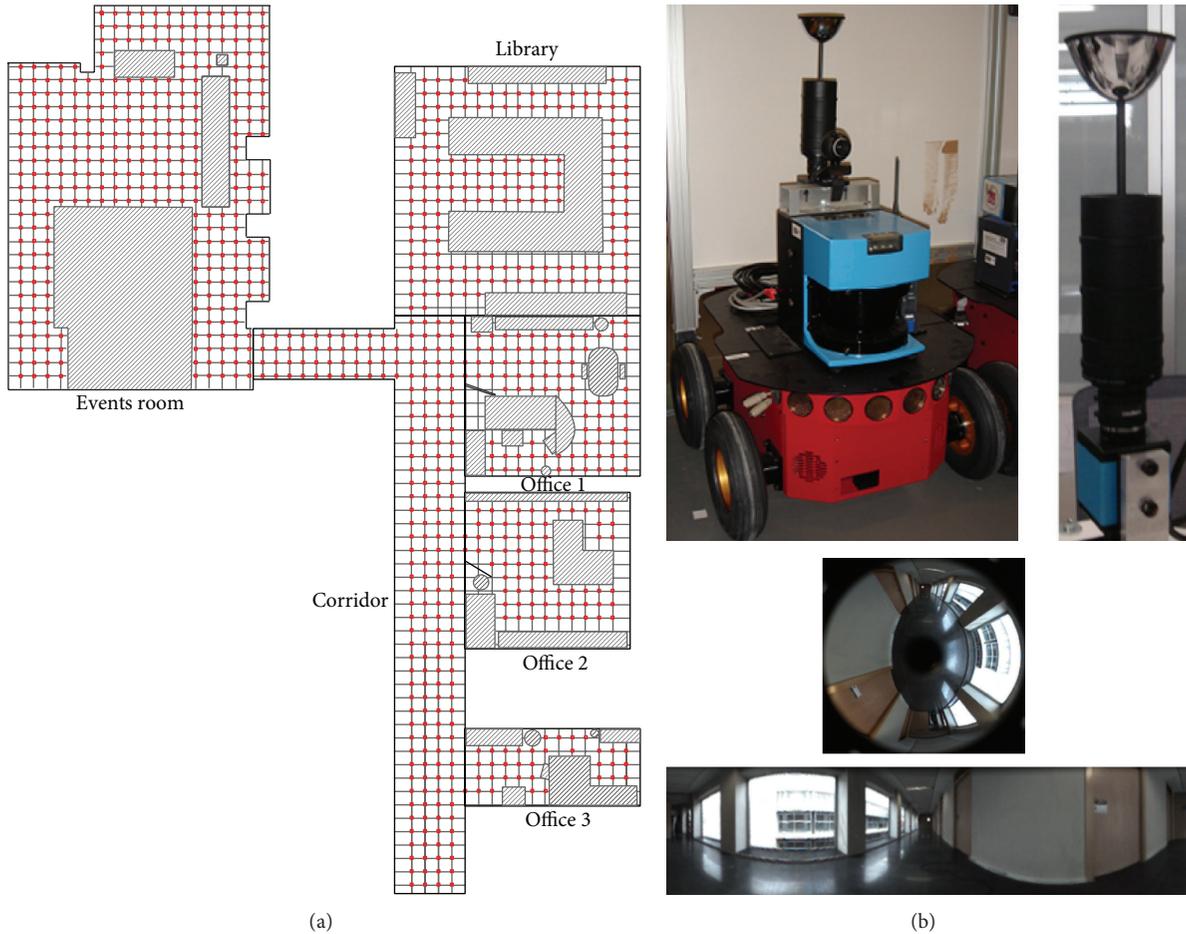


FIGURE 3: (a) Bird's eye view of the environment where set 1 was captured. (b) Catadioptric system mounted on the robot and sample scene captured in the corridor (omnidirectional and panoramic formats).

TABLE 3: Images set 2: rooms considered in the experiments and main parameters.

Room	Number of images	Grid	Size	Resolution
Laboratory	170	30 × 30 cm.	3 × 5 m	81 × 561 pixels
Corridor	200	50 × 50 cm.	5 × 10 m	81 × 561 pixels
Kitchen	108	10 × 10 cm.	1.2 × 1.0 m	81 × 583 pixels
Hall	242	10 × 10 cm.	2.2 × 1.2 m	81 × 583 pixels

flat. They were captured with a camera *ImagingSource DFK 4303* mounted on the robot *ActivMedia Pioneer 3-DX*. The hyperbolic mirror is the model *Accowle Wide View*. This is an interesting database because it presents different grid sizes in each room. It permits testing how this parameter influences the performance of the methods. Table 3 shows the rooms we have used and the main features of the images.

**4.2. Computational Cost.** Previously, Section 3.1 has outlined the contents of the map nodes. Now, the objective of this section consists in making a comparative evaluation of the

computational cost of the four description methods during the creation of the map nodes. This study will be carried out depending on the value of the most relevant parameters of each description method. Data set 1 is used to carry out this comparative evaluation. This is an interesting study as it allows us to know which algorithms could work in real time.

First, Figure 4 shows the computation time using (a) FS versus  $k_1$  and  $k_2$ , and (b) rotational PCA versus  $N_R$ . Second, Figure 5 shows the time when using HOG versus  $k_4$ ,  $l_1$ , and  $d_1$ . At last, Figure 6 shows *gist* with  $m = 8$  versus  $k_6$ ,  $l_2$ , and  $d_2$ . In all cases, the time per image is depicted. The total time to build the map can be obtained by multiplying by 873 (number of images in set 1).

In the case of FS, as  $k_1$  and  $k_2$  increase, the time increases slightly. The cost to obtain the DFT of each row is the same, the difference is in the need of computing the magnitude and argument of a different number of components, which implies a low computational cost. In any case, the computational cost of FS is very low.

As far as rotational PCA is concerned, Figure 4 shows how the time increases exponentially as  $N_R$  does, arriving at up to 110 seconds per image (27 hours to build the whole

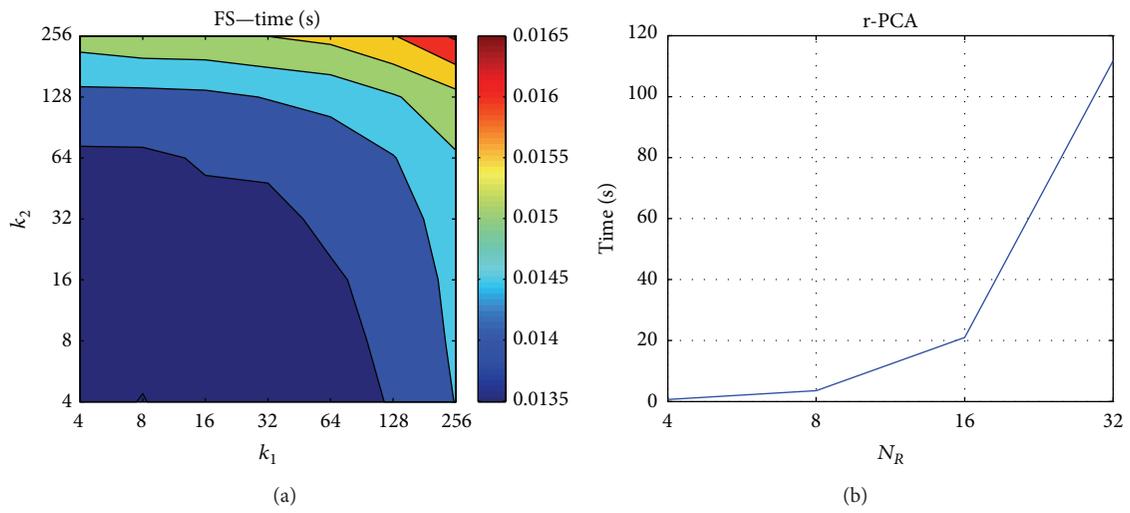


FIGURE 4: Computational cost to obtain the nodes' descriptors using (a) Fourier Signature and (b) rotational PCA.

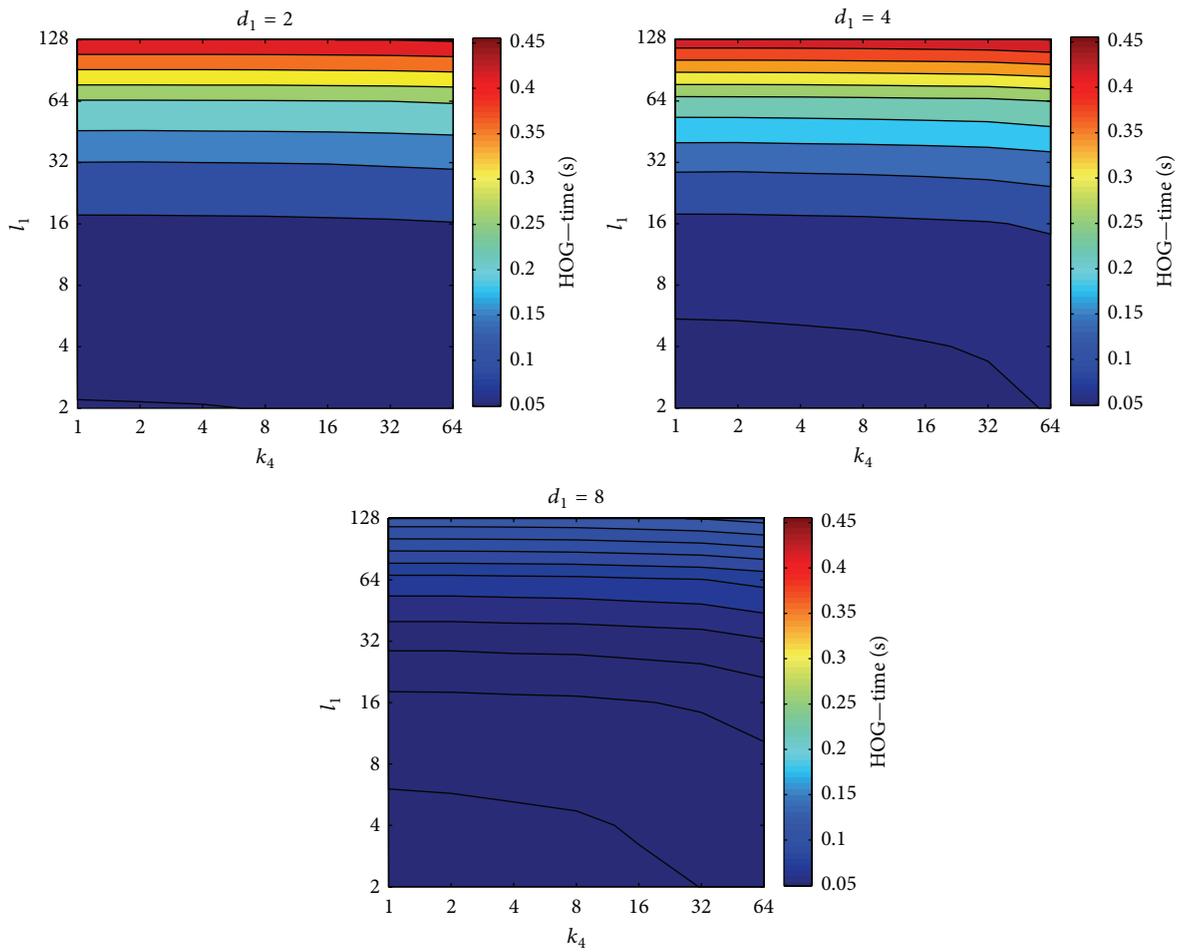


FIGURE 5: Computational cost to obtain the nodes' descriptors using HOG.

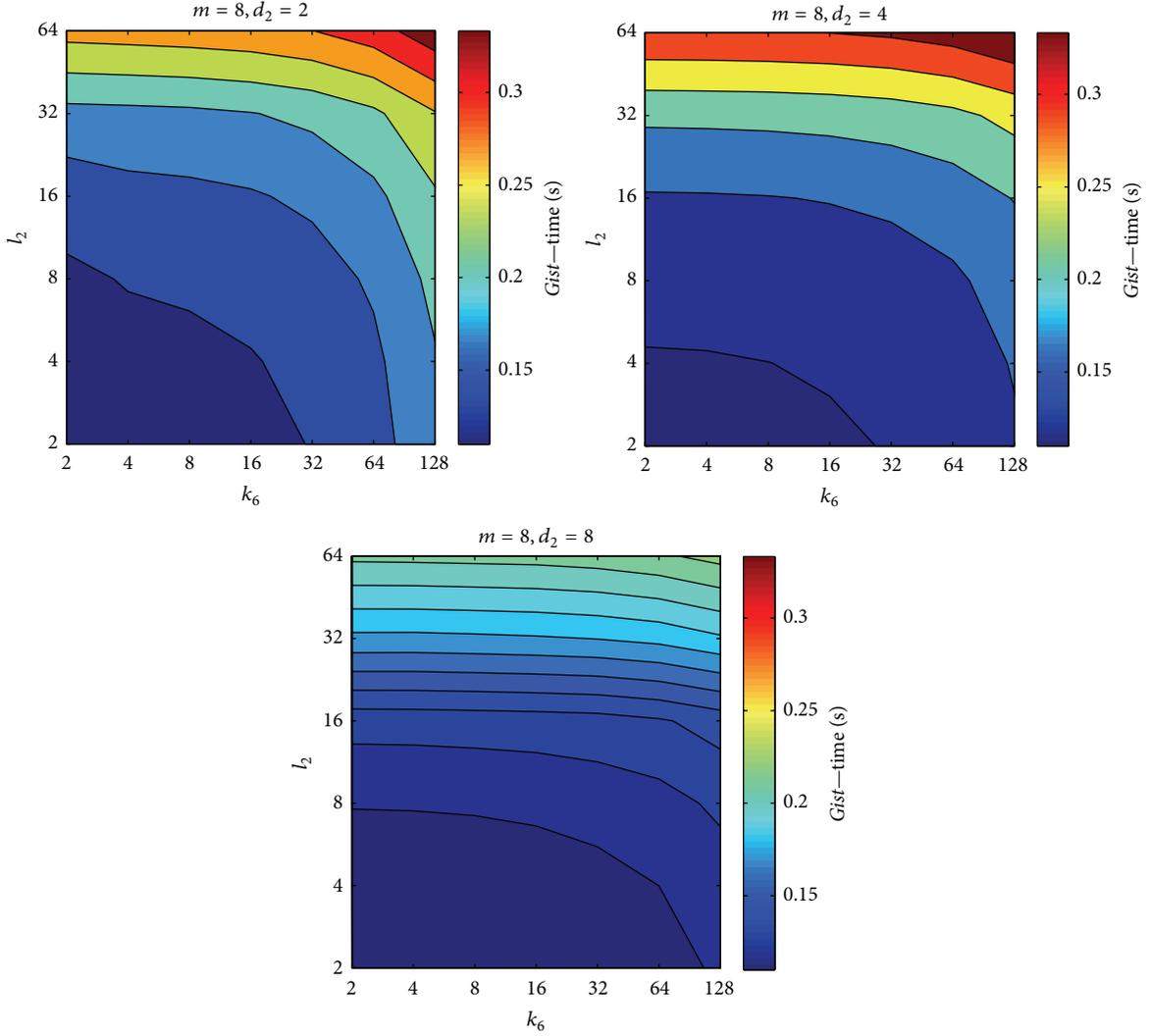


FIGURE 6: Computational cost to obtain the nodes' descriptors using *gist*.

map), when  $N_R = 32$  rotations. It has been impossible to consider a higher number of rotations due to the enormous requirements of memory during the process.

If we analyse now HOG, on the one hand, the influence of  $k_4$  is low and, on the other hand, time increases linearly when  $l_1$  does. At last, when  $d_1$  increases, the time decreases as fewer vertical cells are considered. In general, HOG presents a substantially higher computational cost compared to FS; despite it, the algorithm is quick enough to permit carrying out the mapping process in real time, as the robot explores the unknown environment.

At last, the computational cost of *gist* is, in general, approximately 10 times the cost of FS and similar to HOG. All of FS, HOG, and *gist* are computationally feasible algorithms. Nevertheless, rotational PCA could only be used if the mapping process is allowed to be done offline. Also, the maximum number of rotations included in the map has been  $N_R = 32$ . This means that the resolution in orientation estimation will be low. Anyway, even though the computational cost of rotational PCA had been low enough, this algorithm

would not have permitted building maps online since all the training images must be available to start the process (unless any incremental PCA algorithm is used [32], which would add more computational cost to the process and make it unbearable in real time). The other three algorithms do not present this disadvantage since they are inherently incremental methods (each image is described independently on the rest of images so the robot can build the map as it is exploring the unknown environment).

**4.3. Image Distance versus Geometric Distance.** Once we know the computational cost of the description methods, the objective of this section consists in carrying out several experiments to test the applicability of these methods to the creation of topological maps.

The first experiment consists in studying the relationship between the geometrical distance between the positions where two images have been captured and the distance between the descriptors of these two images. The behaviour of this distance should be monotonically increasing and

linear, at least in a close interval around the point where the reference image was captured.

To carry out this study, several distance measures are taken into consideration. First, these distances are formalized. If we have two descriptors  $\vec{r} \in \mathbb{R}^{l \times 1}$  and  $\vec{s} \in \mathbb{R}^{l \times 1}$ , where  $r_i$  and  $s_i$  are the  $i$ th components of  $\vec{r}$  and  $\vec{s}$ , with  $i = 1, \dots, l$ . The distance between these descriptors can be defined as follows.

(a) *Weighted Metric Distance.* Consider

$$\text{dist}_p(\vec{r}, \vec{s}) = \left( \sum_{i=1}^l \omega_i \cdot |r_i - s_i|^p \right)^{1/p}. \quad (7)$$

If we consider  $\omega_i = 1$ ,  $i = 1, \dots, l$ , the Minkowski distance is obtained. Two particular cases will be considered:  $\text{dist}_1$  (*cityblock* distance), which is defined from the Minkowski distance with  $p = 1$ , and  $\text{dist}_2$  (Euclidean distance), doing  $p = 2$ .

(b) *Pearson Correlation Coefficient.* It is a similitude coefficient that can be obtained as follows:

$$\text{sim}_{\text{pea}}(\vec{r}, \vec{s}) = \frac{\vec{r}_d^T \cdot \vec{s}_d}{|\vec{r}_d| |\vec{s}_d|}, \quad (8)$$

where  $\vec{r}_d = [r_1 - \bar{r}, \dots, r_l - \bar{r}]$  and  $\vec{s}_d = [s_1 - \bar{s}, \dots, s_l - \bar{s}]$ ,  $\bar{r} = (1/l) \sum_j r_j$ ,  $\bar{s} = (1/l) \sum_j s_j$ . It takes values in the range  $[-1, +1]$ . From this similitude coefficient, a distance measure can be defined as follows:

$$\text{dist}_{\text{pea}}(\vec{r}, \vec{s}) = 1 - \text{sim}_{\text{pea}}(\vec{r}, \vec{s}). \quad (9)$$

(c) *Inner Product.* It is also a similitude coefficient that can be calculated as the scalar product between the two vectors to compare

$$\text{sim}_{\text{cos}}(\vec{r}, \vec{s}) = \frac{\vec{r}^T \cdot \vec{s}}{|\vec{r}| |\vec{s}|}. \quad (10)$$

As shown in the equation,  $\vec{r}$  and  $\vec{s}$  are usually normalized. In this case, this measure is known as *cosine similitude* and takes values in the range  $[-1, +1]$ . The corresponding distance value is

$$\text{dist}_{\text{cos}}(\vec{r}, \vec{s}) = 1 - \text{sim}_{\text{in}}(\vec{r}, \vec{s}). \quad (11)$$

(d) *Other Distance Measures.* Other distance measures have been considered in the study, as they have provided good results when applied to very-high dimensional data in clustering tasks [33]. We name them log and root distances:

$$\begin{aligned} \text{dist}_{\log}(\vec{r}, \vec{s}) \\ = -\log_{10} \left( 1 - \frac{1}{l} \sum_{i=1}^l \frac{|r_i - s_i|}{\max(r_i) - \min(r_i)} \right), \end{aligned} \quad (12)$$

where  $\max(r_j)$  and  $\min(r_j)$  are, respectively, the maximum and minimum value among the  $j$  components of the  $n$  vectors in  $\mathbf{R} = \{\vec{r}_1, \dots, \vec{r}_n\}$ . This way, the distance does not only depend on  $\vec{r}$  and  $\vec{s}$ , but also on the set of vectors in  $\mathbf{R}$ :

$$\text{dist}_{\text{root}}(\vec{r}, \vec{s}) = \sqrt{\frac{1}{l} \sum_{i=1}^l \left( \frac{r_i - s_i}{r_i + s_i} \right)^2}. \quad (13)$$

To study the relation between the image distance (distance between the descriptors of two images) and the geometric distance (Euclidean distance between the points where these images were captured) the rooms *kitchen* and *hall* of data set 2 have been used, since these are the two rooms whose grid presents a higher resolution ( $10 \times 10$  cm). In both cases, from a reference point, some sets of scenes both horizontally and vertically have been taken and the distance between the reference image and all of them has been obtained. The next figures show the results obtained (average distance and variance) after this set of experiments.

First, Figure 7 shows the distance results when using FS. This figures show how, in this case, the different distance measures present quite similar results. In a close interval to the reference image, the image distance increases (quite linearly in the case of the correlation and cosine distances). However, they present a nondesirable behaviour since they reach a maximum and then they begin to decrease. The cosine distance is not shown as it provides a very similar result to the correlation.

Next, Figure 8 shows the results when using rotational PCA to describe scenes. In all cases,  $k_3 = 200$  components have been used. The result obtained with the distance *cityblock* is remarkable because, despite being the simplest measure, it behaves quite linearly when the number of rotations is high enough (but it presents a local minimum in the middle). Logarithm and root distances present also relatively good results. The data in Figure 9 allow us to analyse the influence of the number of PCA components. In all cases, including a low number of components (very compact descriptors), the behaviour is quite linear and monotonous with some distance measures.

Thirdly, Figure 10 shows the results obtained when the images are described through HOG. In all cases, the results are quite similar to the FS. However, the local maximum is reached in a closer point to the reference image. This fact limits the validity range of the computed distance.

To finish the distance results, we show the results obtained with *gist* in Figure 11. In this case, thanks to the linearity and monotony, the results obtained with correlation (and cosine) must be highlighted.

As a final conclusion, the FS and HOG descriptors present a limited utility to estimate the topological distance between images, provided that the behaviour of the distances is not monotonous (FS presents a larger useful interval). Rotational PCA presents a relatively good behaviour when using *cityblock*, Euclidean, and Minkowski distance. At last, the excellent performance of *gist* with the *cityblock* and correlation distances must be highlighted, due to their monotony and linearity. The goodness of this configuration suggests

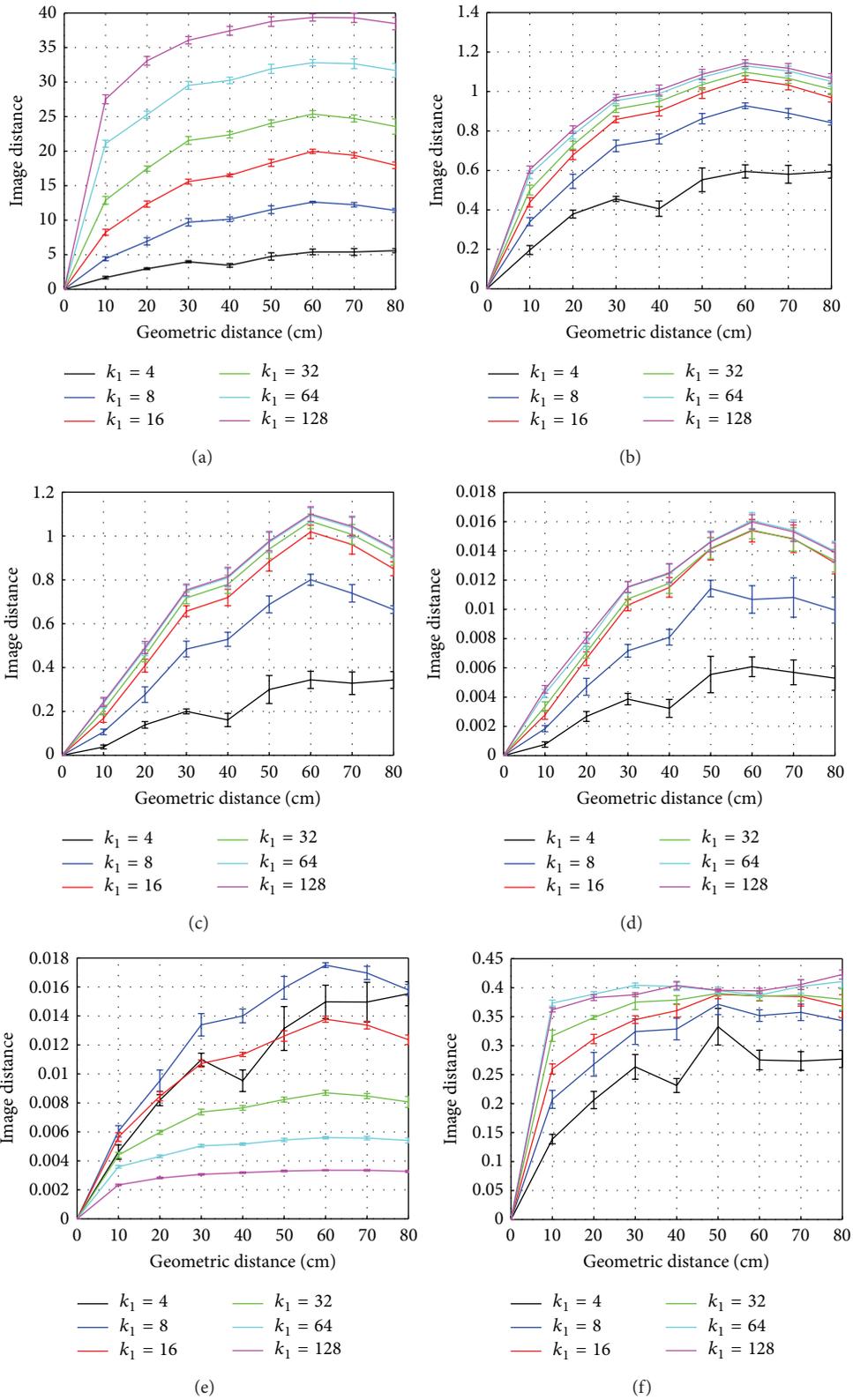


FIGURE 7: FS. Image distance versus geometric distance, depending on  $k_1$ . Distance measure: (a) *cityblock*, (b) Euclidean, (c) weighted  $p = 3$ ,  $\omega_i = 0.95^i$ , (d) correlation, (e) logarithm, and (f) root.

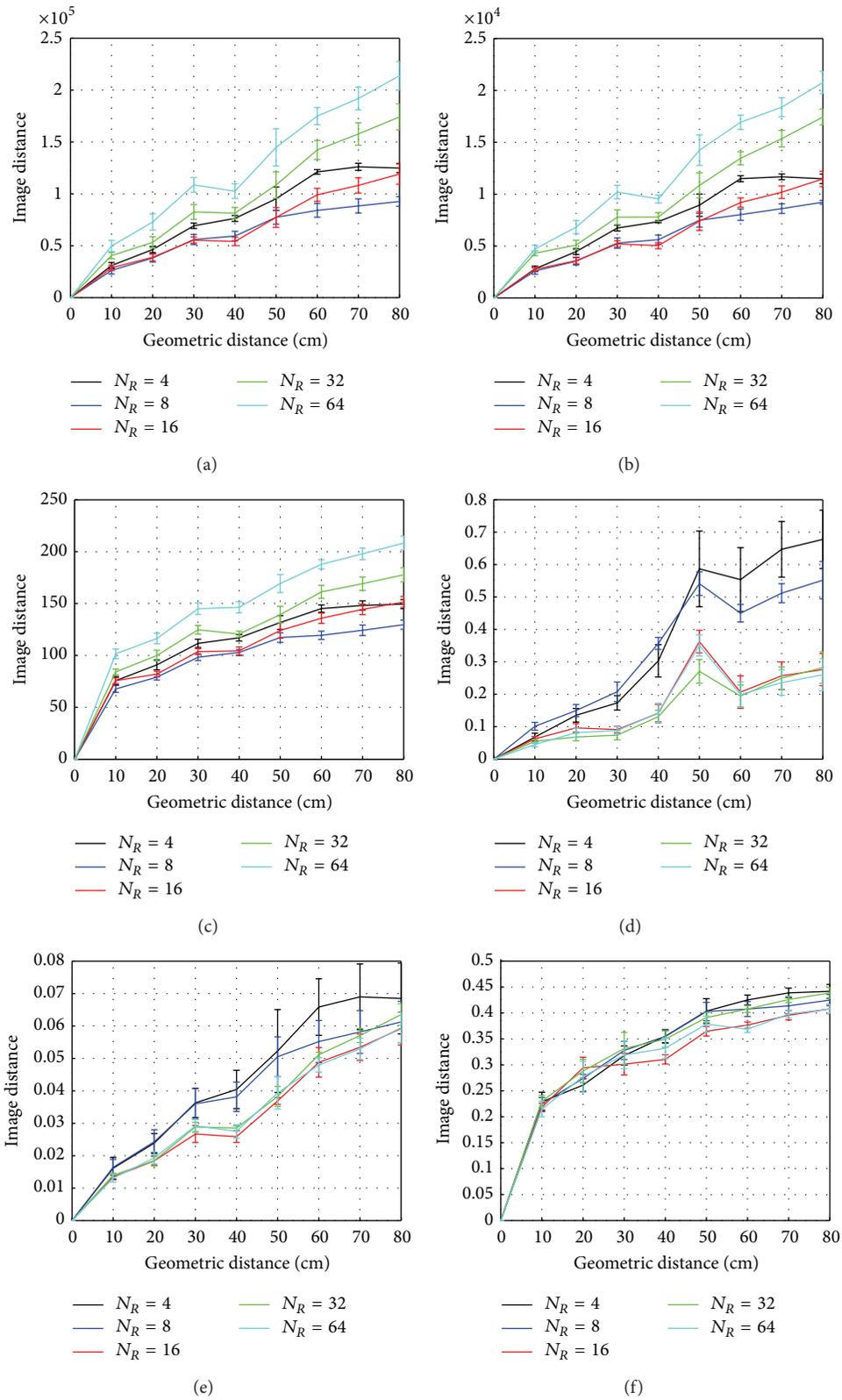


FIGURE 8: Rotational PCA. Image distance versus geometric distance, depending on  $N_R$ . Distance measure: (a) *cityblock*, (b) Euclidean, (c) weighted  $p = 2, \omega_i = 0.95^i$ , (d) correlation, (e) logarithm, and (f) root.

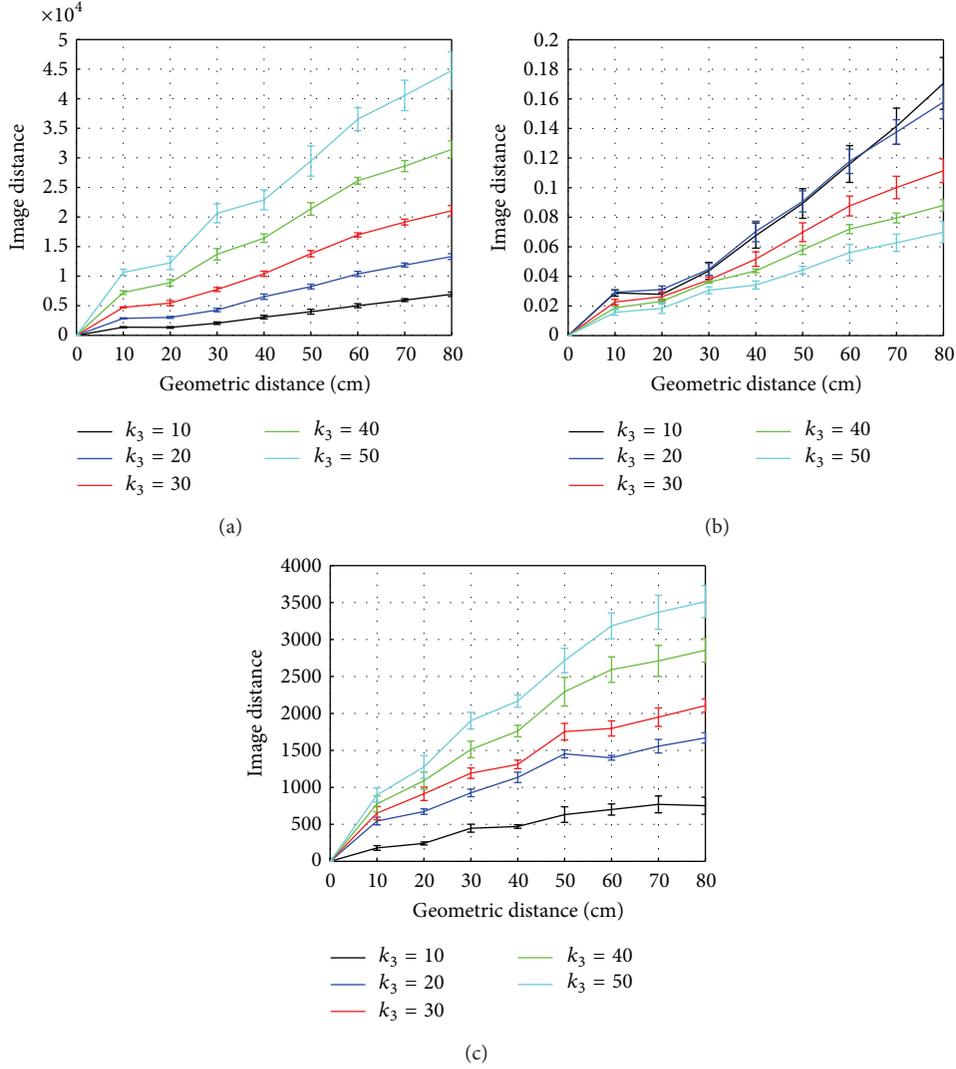


FIGURE 9: Rotational PCA. Image distance versus geometric distance, depending on  $k_3$ . (a) cityblock distance and  $N_R = 64$ , (b) logarithm distance and  $N_R = 64$ , and (c) Euclidean distance and  $N_R = 4$ .

that it could be the first option to implement a topological mapping algorithm.

**4.4. Topological Model.** This section reflects the last experiment carried out. The algorithm presented in Section 3.2 has been used to build several topological maps using the data set 2. This data set presents different grid steps, depending on the room considered. This way, it allows us to study the influence of this important parameter.

As far as the configuration of the mass-spring-damper algorithm is concerned, the most critical parameter is the spring constant. If we consider that all the springs have the same elastic constant, the results are not consistent, because the presence of *visual aliasing* in the environment introduces undesired forces in the system. To avoid this effect, each elastic constant is calculated depending on the distances

between the descriptors of the two particles  $i$  and  $j$  linked by this spring, according to the following expression:

$$k_{ij} = \min \left( \frac{c}{\text{dist}(\vec{d}_i, \vec{d}_j)^2}, 100 \right), \quad (14)$$

$$i, j = 1, \dots, n, \quad i \neq j,$$

where  $c$  is the average slope measured on Figures 7–11, depending on the selected descriptor and parameters. The value of  $k_{ij}$  has been limited to 100 to avoid the presence of very high efforts.

At last, the natural length of each spring is equal to the distance between the descriptors of the particles linked by the spring:

$$l_{ij}^0 = \text{dist}(\vec{d}_i, \vec{d}_j). \quad (15)$$

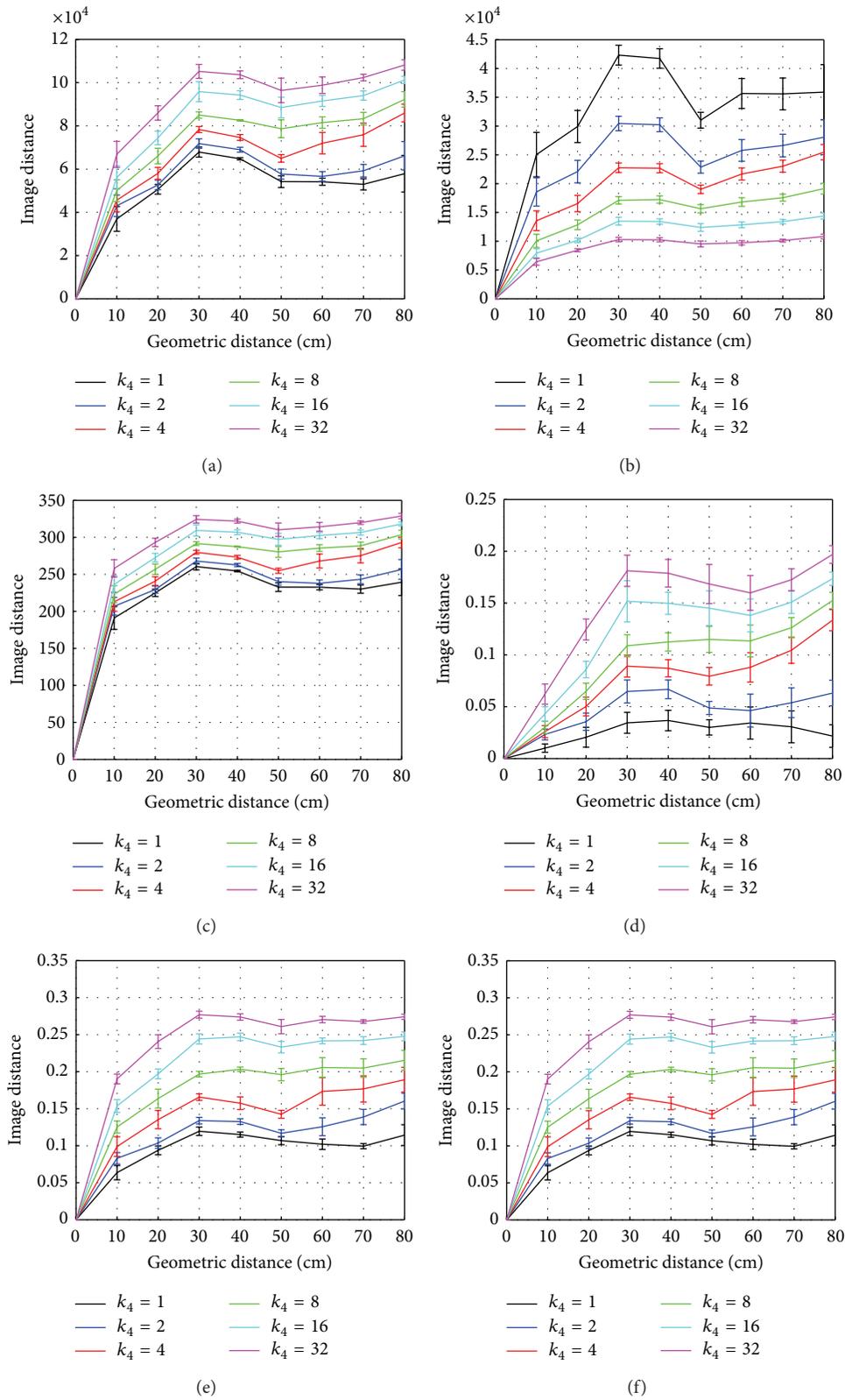


FIGURE 10: HOG. Image distance versus geometric distance, depending on  $k_4$ . Distance measure: (a) *cityblock*, (b) Euclidean, (c) weighted  $p = 3$ ,  $\omega_i = 1$ , (d) correlation, (e) logarithm, and (f) root.

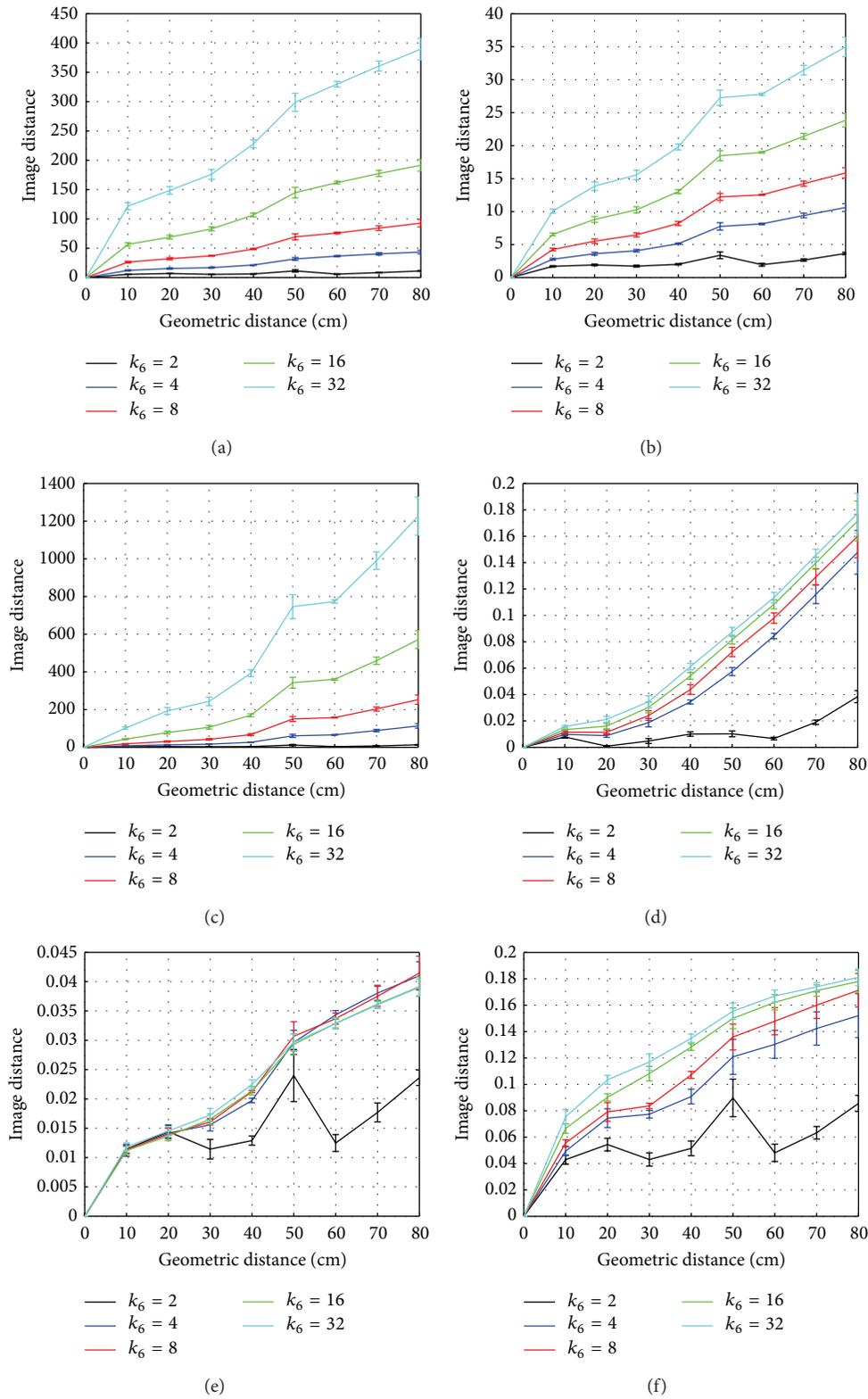


FIGURE 11: *Gist*. Image distance versus geometric distance, depending on  $k_6$ . Distance measure: (a) *cityblock*, (b) Euclidean, (c) weighted  $p = 3$ ,  $\omega_i = 1$ , (d) correlation, (e) logarithm, and (f) root.

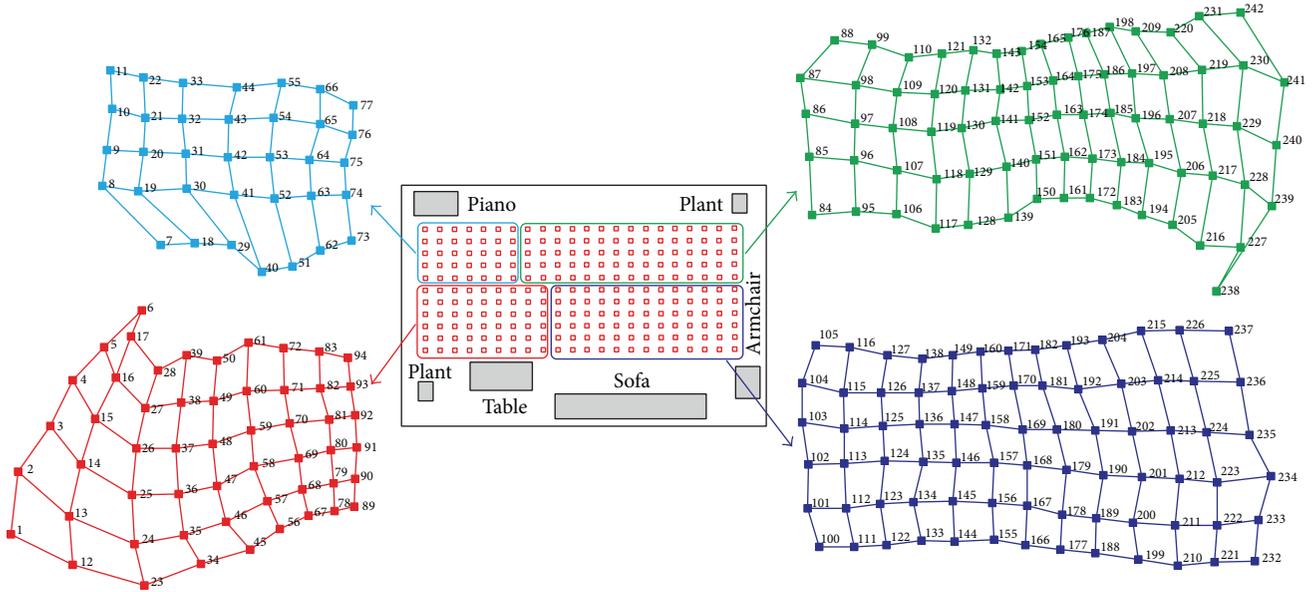


FIGURE 12: Topological maps created in the *hall*, data set 2. The grid size is  $10 \times 10$  cm.

To finish, all the particles are considered to have the same mass  $m_i = 1$ ,  $i = 1, \dots, n$ , since our experiments have shown that it is not a relevant parameter. The damping constant of all the dampers is set to  $\kappa_{ij} = 0.6$ . Thanks to this dynamic friction, the behaviour of the system tends to be overdamped and more stable, permitting a gradual evolution from the initial position to the steady state, without large oscillations. To finish, we have defined the time step  $\Delta t = 0.03$  s. It is an important parameter that influences both the settling time and the stability of the resulting system. A low value supposes a high settling time and a high value makes this time lower but the movement between two consecutive iterations may be so high that the system could destabilize.

After a complete bank of experiments, the best results have been obtained with the *gist* descriptor with  $k_6 = 16$  blocks,  $m = 16$  orientations, and correlation distance and with the FS descriptor with  $k_1 = 32$  blocks and correlation distance. These results are in line with Section 4.3. HOG has not provided good results, as Figure 10 suggested.

Figures 12, 13, and 14 show some of the topological maps created in three different rooms of data set 2 (*hall*, *laboratory*, and *corridor*, resp.). We show the results of these rooms because they have different grid size. In each room, several sets of images, with different size and distribution along all the space of the room, have been chosen. Then, the mapping algorithm has been applied. The final distribution of each map is shown. In these maps, the lines are drawn just with representative purposes (when the algorithm starts, it has no information about the initial positions nor about the vicinity relations).

The figures show that, despite the different grid size, relatively good results are achieved in all cases. This way, global-appearance descriptors prove to be a good choice for the creation of topological maps where the concepts of closeness and farness are included.

Comparing to feature-based techniques, in a previous work [34], a new global-appearance description method was proposed and a preliminary comparison with a classical global-appearance method (the Fourier Signature) and a feature-based method (SIFT features) was carried out. The results showed that global-appearance descriptors are robust to solve the localization process and their computational cost is relatively low, improving the performance of local feature descriptors.

To finish, Table 4 shows a final comparison of the performance of the four methods in mapping tasks. First, to compare the computational cost, the table shows the minimum and the maximum necessary time ( $t_{\min}$  and  $t_{\max}$ , resp.) to include each image in the model. Second, to study the relation between the image distance and the geometric distance, a least squares linear fit has been carried out with all the curves in Figures 7–11. In all cases, the origin has been weighted to ensure that the resulting line passes through it. The table shows the results of the best fit: the slope  $m$ , the coefficient  $r^2$ , and the values of the parameters.

## 5. Conclusion and Future Works

This paper has focused on the study of the mapping problem. It has been addressed from a topological point of view, using the information provided by an omnidirectional vision sensor to build the model, and methods based on global appearance to extract relevant information from the scenes. The work has carried out a comparative evaluation between some renowned description methods in map building tasks.

The main contributions of the paper include an exhaustive study of visual appearance techniques (FS, PCA, HOG, and *gist*) and the adaptation of some of these algorithms to store position and orientation information from panoramic scenes. Also, the computational cost to build the nodes

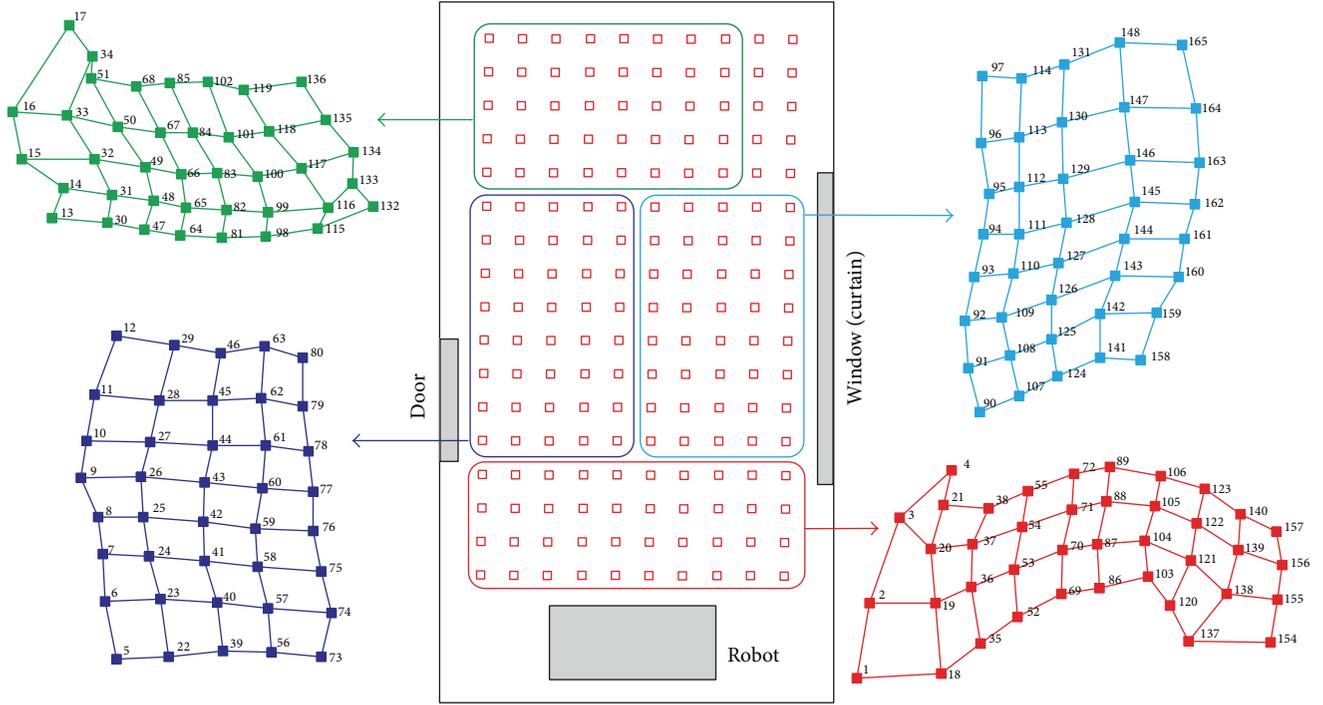


FIGURE 13: Topological maps created in the *laboratory*, data set 2. The grid size is  $30 \times 30$  cm.

TABLE 4: Performance of the description methods: computational cost per image to build the model and best linear fit of the image distance versus the geometric distance.

Descriptor	Time		$m$	$r^2$	Best linear fit	
	$t_{\min}$ (ms)	$t_{\max}$ (ms)			Parameters	Distance
FS	13	17	0.50	0.9734	$k_1 = 4$	Weighted
r-PCA	675	111740	$2.7 \cdot 10^5$	0.9856	$k_3 = 50, N_R = 64$	<i>Cityblock</i>
HOG	49	456	0.17	0.9576	$k_4 = 16$	Correlation
<i>Gist</i>	110	333	14	0.9902	$k_6 = 4$	Correlation

of the map has been studied, including the influence of the most relevant parameters. This study has revealed that FS, HOG, and *gist* present a reasonable computational cost and, from this point of view, their use could be feasible in real time applications. Besides this, the performance of the descriptors has been tested in mapping tasks. First, we have focused on the relation between the image distance and the geometric distance, which allows us to know the descriptors that best reflect an idea of closeness and farness, since they are two important concepts to reflect in the map. All the description methods have been tested along with several distance measures, and the results have shown that *gist* and FS descriptors with certain distance measures present positive results. Second, a mass-spring-damper method has been implemented to build topological maps, their parameters have been tuned and several experiments have been carried out. To finish, several topological maps have been built, including not only connectivity but also closeness and farness concepts. The results have shown the goodness of the mapping approach and of the parameters tuning.

These results have demonstrated that global-appearance methods are a feasible approach to solve the mapping task. Thanks to them, the robot can build a model of the environment that goes beyond the classical *topological maps* since the model is a version of the original grid except for a scale factor. This suggests that the model could be used to estimate with accuracy the position and orientation of the robot in the environment, with computational efficiency. This fact may have interesting implications in future developments in the field of mobile robotics. As an example, this concept can be used to build hybrid maps that arrange the information in several layers, with different accuracy: a high level layer that permits carrying out a rough and quick localization and a lower layer that contains information with geometric accuracy and allows the robot to refine the estimation of its position. Global-appearance methods can be used on their own or in conjunction with feature-based techniques to develop algorithms that face these problems efficiently.

All these facts encourage us to go into this framework in depth. To build a fully autonomous mapping and localization

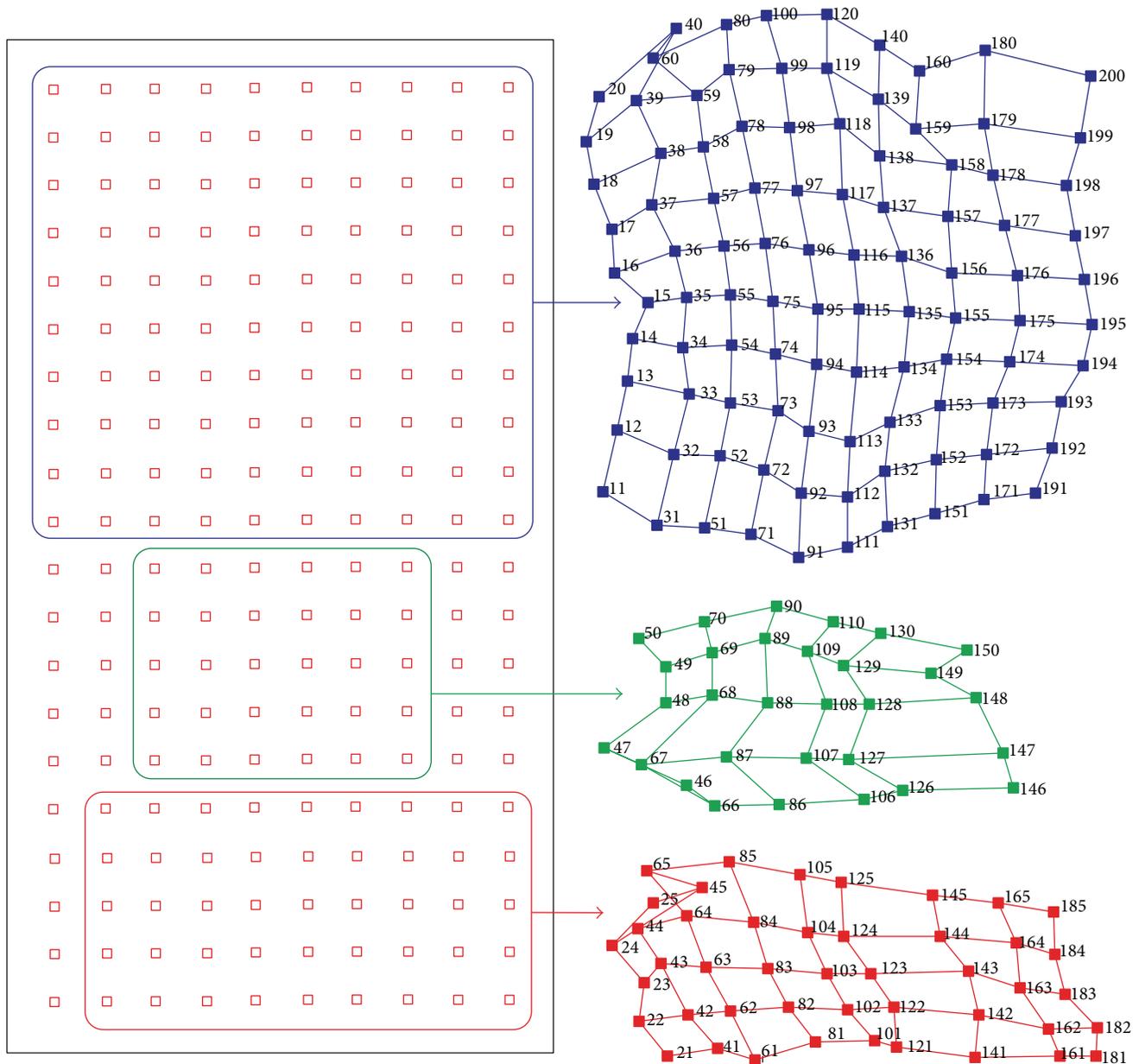


FIGURE 14: Topological maps created in the *corridor*, data set 2. The grid size is  $50 \times 50$  cm.

system several future works should be considered. First, the image collection process could be automated to obtain an optimal representation of the environment. Second, this model must be used to estimate the current position and orientation of the robot taking into account typical situations such as changes in lighting conditions or visual occlusions. At last, both processes could be integrated in a topological SLAM system that carries out both the model creation and the localization from the scratch. To optimize these algorithms we also consider carrying out a complete comparison between global-appearance and feature-based techniques as a future work.

## Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

## Acknowledgments

This work has been supported by the Spanish Government through the project DPI 2013-41557-P, *Navegación de Robots en Entornos Dinámicos Mediante Mapas Compactos con Información Visual de Apariencia Global*, and by the Generalitat Valenciana through the project GV/2015/031: *Creación de*

*Mapas Topológicos a Partir de la Apariencia Global de un Conjunto de Escenas.*

## References

- [1] J. Gaspar, N. Winters, and J. Santos-Victor, "Vision-based navigation and environmental representations with an omnidirectional camera," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 890–898, 2000.
- [2] S. Thrun, "Robotic mapping: a survey, in exploring artificial intelligence," in *The New Milenium*, pp. 1–35, Morgan Kaufmann, San Francisco, Calif, USA, 2003.
- [3] A. Gil, Ó. Reinoso, M. Ballesta, M. Juliá, and L. Payá, "Estimation of visual maps with a robot network equipped with vision sensors," *Sensors*, vol. 10, no. 5, pp. 5209–5232, 2010.
- [4] E. Garcia-Fidalgo and A. Ortiz, "Vision-based topological mapping and localization methods: a survey," *Robotics and Autonomous Systems*, vol. 64, pp. 1–20, 2015.
- [5] K. Konolige, E. Marder-Eppstein, and B. Marthi, "Navigation in hybrid metric-topological maps," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 3041–3047, Shanghai, China, May 2011.
- [6] A. Angeli, S. Doncieux, J.-A. Meyer, and D. Filliat, "Visual topological slam and global localization," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 4300–4305, IEEE, Kobe, Japan, May 2009.
- [7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [8] C. Valgren and A. J. Lilienthal, "SIFT, SURF & seasons: appearance-based long-term localization in outdoor environments," *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 149–156, 2010.
- [9] A. C. Murillo, J. J. Guerrero, and C. Sagüés, "SURF features for efficient robot localization with omnidirectional images," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 3901–3907, IEEE, Rome, Italy, April 2007.
- [10] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: speeded up robust features," in *Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006. Proceedings, Part I*, A. Leonardis, H. Bischof, and A. Pinz, Eds., vol. 3951 of *Lecture Notes in Computer Science*, pp. 404–417, Springer, Berlin, Germany, 2006.
- [11] J. Kosecka, L. Zhou, P. Barber, and Z. Duric, "Qualitative image based localization in indoors environments," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. II-3–II-8, Madison, Wis, USA, June 2003.
- [12] E. Menegatti, T. Maeda, and H. Ishiguro, "Image-based memory for robot navigation using properties of omnidirectional images," *Robotics and Autonomous Systems*, vol. 47, no. 4, pp. 251–267, 2004.
- [13] F. Rossi, A. Ranganathan, F. Dellaert, and E. Menegatti, "Toward topological localization with spherical fourier transform and uncalibrated camera in," in *Proceedings of the International Conference on Simulation, Modeling and Programming for Autonomous Robots (SIMPAN '08)*, pp. 319–350, Springer, Venice, Italy, 2008.
- [14] A. Gil, O. M. Mozos, M. Ballesta, and O. Reinoso, "A comparative evaluation of interest point detectors and local descriptors for visual SLAM," *Machine Vision and Applications*, vol. 21, no. 6, pp. 905–920, 2010.
- [15] A. Guérin-Dugué and A. Oliva, "Classification of scene photographs from local orientations features," *Pattern Recognition Letters*, vol. 21, no. 13-14, pp. 1135–1140, 2000.
- [16] M. Kirby, *Geometric Data Analysis: An Empirical Approach to Dimensionality Reduction and the Study of Patterns*, Wiley-Interscience, 2001, <http://books.google.es/books?id=nRmFQgAACAAJ>.
- [17] A. Leonardis and H. Bischof, "Robust recognition using eigenimages," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 99–118, 2000.
- [18] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [19] M. Jogan and A. Leonardis, "Robust localization using eigenspace of spinning-images," in *Proceedings of the IEEE Workshop on Omnidirectional Vision*, pp. 37–44, IEEE, Hilton Head Island, SC, USA, June 2000.
- [20] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, San Diego, Calif, USA, June 2005.
- [21] Q. Zhu, S. Avidan, M.-C. Yeh, and K.-T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 2, pp. 1491–1498, IEEE, June 2006.
- [22] M. Hofmeister, M. Liebsch, and A. Zell, "Visual self-localization for small mobile robots with weighted gradient orientation histograms," in *Proceedings of the 40th International Symposium on Robotics (ISR '09)*, pp. 87–91, IFR, Barcelona, Spain, March 2009.
- [23] L. Payá, F. Amorós, L. Fernández, and O. Reinoso, "Performance of global-appearance descriptors in map building and localization using omnidirectional vision," *Sensors*, vol. 14, no. 2, pp. 3033–3064, 2014.
- [24] A. Oliva and A. Torralba, "Building the gist of ascene: the role of global image features in recognition," *Progress in Brain Reasearch*, vol. 155, pp. 23–36, 2006.
- [25] I. Biederman, "Aspects and extension of a theory of human image understanding," in *Computational Processes in Human Vision: An Interdisciplinary Perspective*, Ablex, Norwood, NJ, USA, 1988.
- [26] C. Siagian and L. Itti, "Biologically inspired mobile robot vision localization," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 861–873, 2009.
- [27] C.-K. Chang, C. Siagian, and L. Itti, "Mobile robot vision navigation & localization using gist and saliency," in *Proceedings of the 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 4147–4154, IEEE, Taipei, Taiwan, October 2010.
- [28] A. C. Murillo, G. Singh, J. Kosecká, and J. J. Guerrero, "Localization in urban environments using a panoramic gist descriptor," *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 146–160, 2013.
- [29] A. Selle, M. Lentine, and R. Fedkiw, "A mass spring model for hair simulation," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 64:1–64:11, 2008.
- [30] Automation-Robotics and Computer Vision Research Group (ARVC), "Quorum 5 set of images," Miguel Hernández University, Elche, Spain, <http://arvc.umh.es/db/images/quorumv/>.

- [31] R. Möller, A. Vardy, S. Kreft, and S. Ruwisch, "Visual homing in environments with anisotropic landmark distribution," *Autonomous Robots*, vol. 23, no. 3, pp. 231–245, 2007.
- [32] M. Artač, M. Jogan, and A. Leonardis, "Mobile robot localization using an incremental eigenspace model," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1025–1030, IEEE, May 2002.
- [33] H. Spat, *Clustering Analysis Algorithms for Data Reduction and Classification of Objects*, Ellis Horwood, New York, NY, USA, 1982.
- [34] Y. Berenguer, L. Payá, M. Ballesta, and O. Reinoso, "Position estimation and local mapping using omnidirectional images and global appearance descriptors," *Sensors*, vol. 15, no. 10, pp. 26368–26395, 2015.

## Research Article

# Optical Flow Sensor/INS/Magnetometer Integrated Navigation System for MAV in GPS-Denied Environment

**Chong Shen,<sup>1</sup> Zesen Bai,<sup>2</sup> Huiliang Cao,<sup>1</sup> Ke Xu,<sup>1</sup> Chenguang Wang,<sup>1</sup> Huaiyu Zhang,<sup>1</sup> Ding Wang,<sup>1</sup> Jun Tang,<sup>1</sup> and Jun Liu<sup>1</sup>**

<sup>1</sup>National Key Laboratory for Electronic Measurement Technology,  
Key Laboratory of Instrumentation Science & Dynamic Measurement, Ministry of Education, School of Instrument and Electronics,  
North University of China, Taiyuan 030051, China

<sup>2</sup>School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China

Correspondence should be addressed to Jun Liu; liuj@nuc.edu.cn

Received 19 November 2015; Accepted 12 January 2016

Academic Editor: Fadi Dornaika

Copyright © 2016 Chong Shen 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.

The drift of inertial navigation system (INS) will lead to large navigation error when a low-cost INS is used in microaerial vehicles (MAV). To overcome the above problem, an INS/optical flow/magnetometer integrated navigation scheme is proposed for GPS-denied environment in this paper. The scheme, which is based on extended Kalman filter, combines INS and optical flow information to estimate the velocity and position of MAV. The gyro, accelerator, and magnetometer information are fused together to estimate the MAV attitude when the MAV is at static state or uniformly moving state; and the gyro only is used to estimate the MAV attitude when the MAV is accelerating or decelerating. The MAV flight data is used to verify the proposed integrated navigation scheme, and the verification results show that the proposed scheme can effectively reduce the errors of navigation parameters and improve navigation precision.

## 1. Introduction

Recently, the autonomous operating unmanned aerial vehicle (MAV) is becoming popular in military and civilian applications [1–3], such as aerial surveillance, cargo delivery, search and rescue, and remote sensing and mapping. A reliable navigation system is very important for safely operating the MAV. Currently, global positioning system (GPS) [4] and inertial navigation system (INS) [5] are the most widely used navigation techniques for the MAV. GPS is based on satellite which can provide relatively consistent accuracy navigation; however, the satellite signals would be lost in city canyon or indoor environments [6, 7]. INS is a self-contained device which can offer a complete set of navigation parameters with high frequency, including attitude, velocity, and position; however, the rapid growth of systematic errors with time is a main drawback of stand-alone INS [8, 9].

Due to the drawbacks of GPS and INS, lots of novel navigation devices have been studied. For example, optical flow sensor has been proved which can provide reliable

velocity and position information when applied for robot localization or navigation [10, 11]. Optical flow techniques are motivated by bird and insect flights which are used to solve the navigation problem. Optical flow can be thought to be the 2D projection of the 3D perceived motion of the objects, which has been widely utilized for motion estimation [12, 13]. When facing the ground, the optical flow sensors can be used for accurate velocity calculation, and the position can be obtained by integration. For example, computer mouse sensor is a successful application of optical flow theory. Benefiting from previous study of optical flow algorithm and hardware design, robotics researchers have employed optical flow sensors for the MAV navigation. Ding et al. [14] added optical flow into the GPS/INS integration for UAV navigation where optical flow rate is used as backup velocity aiding in case of GPS signal outages, and the verification results showed that, by using optical flow, the height estimation was still within a meter range within 20 seconds after GPS dropped out. Mercado et al. [15] proposed a GPS/INS/optical flow data

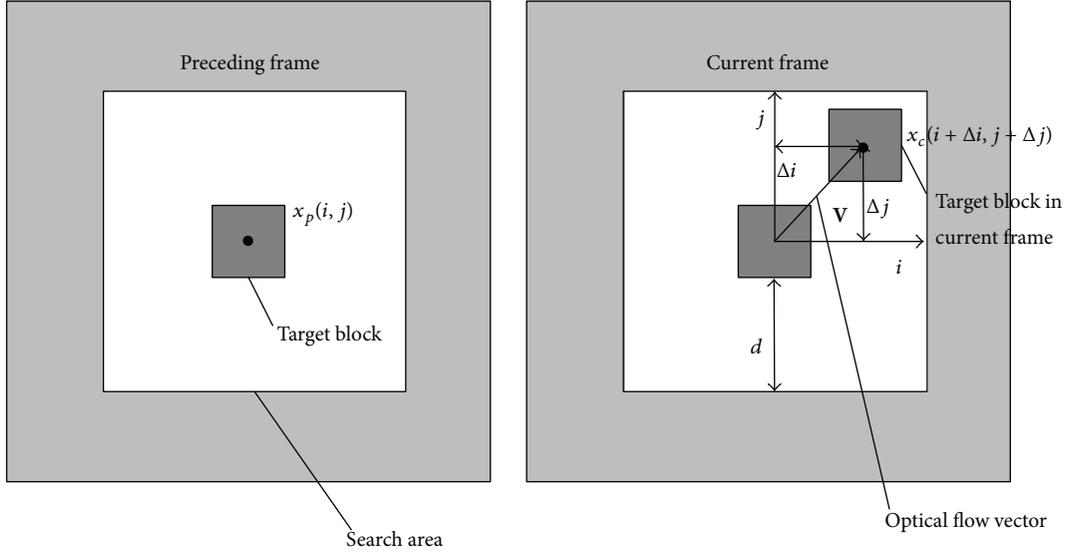


FIGURE 1: Block matching algorithm based on sum of absolute differences.

fusion method for position and velocity estimation; experiment results showed that a good velocity measurement could be obtained by optical flow under rough GPS conditions. Rhudy et al. [16] aided wide-field optical flow to INS for UAV, where the wide-field optical flow was used to regulate INS drift for the purpose of GPS-denied navigation. Gageik et al. [17] applied an optical flow sensor for 2D positioning of UAV, and position hold result showed that the standard deviation of the position error was just 10 cm and the position error was just about 30 cm after landing. It can be clearly seen that the optical flow is an effective solution for velocity and position measurement. However, traditional optical flow is detected by video camera, which is not suitable for MAV because of the weight, size, and power. In 2013, an optical flow sensor based on a machine vision CMOS image sensor for MAV was proposed and named as PX4FLOW [18], and the indoor verification results showed that the error was just 0.5 m with the overall trajectory being 28.44 m. Compared to traditional optical flow sensors, PX4FLOW is of low power, low latency, small size, and low cost; therefore, it is very suitable for MAV applications.

In this paper, an optical flow sensor/INS/magnetometer integrated navigation system is proposed for MAV, where optical flow sensor is used to measure the velocity and position of the MAV, and the magnetometer is used to measure the attitude of the MAV; then, these navigation parameters are applied to calibrate the drift of INS by using an extend Kalman filter. This paper is organized as follows: Section 2 is the theory of velocity and position estimation by optical flow. The data fusion model is given in Section 3. Section 4 is the experiment and verification. Section 5 is the conclusion.

## 2. Estimation of Velocity and Position by Optical Flow Sensor

*2.1. The Calculation of Optical Flow.* Optical flow is the projection of 3D relative motion onto a 2D image plane. In order

to calculate the optical flow, the researchers have proposed many solutions, such as Lucas-Kanade algorithm, Horn-Schunck algorithm, image interpolation algorithm, block matching algorithm, and feature matching algorithm. In our study, considering the complexity of the software calculation and hardware platform, the block matching algorithm (BMA) based on minimum mean absolute error (MMAE) and the sum of absolute differences (SAD) are selected to calculate the optical flow, which is shown in Figure 1.

As shown in Figure 1,  $x_p(i, j)$  is set as the gray value of  $n \times n$  target block selected from the previous frame, and  $x_c(i + \Delta i, j + \Delta j)$  is the gray value of the target block which is to be matched in the searching area of current frame, where  $1 \leq i, j \leq n$ ,  $-d \leq \Delta i, \Delta j \leq d$ . The principle of BMA based on MMAE and the SAD is to search  $\Delta i$  and  $\Delta j$  which satisfy (1); then, the optical flow vector  $\mathbf{V} = r(\Delta i, \Delta j)^T$  can be obtained, where the unit of  $\mathbf{V}$  is pixel/sec;  $r$  is the sample frequency of the camera and the unit is frame/e/sec;  $U$  is the minimum mean absolute error:

$$\text{SAD}(\Delta i, \Delta j) = \sum_{j=1}^n \sum_{i=1}^n |x_c(i + \Delta i, j + \Delta j) - x_p(i, j)|, \quad (1)$$

$$U = \min_{(\Delta i, \Delta j)} \{\text{SAD}(\Delta i, \Delta j)\}, \quad (2)$$

$$\mathbf{V} = r(\Delta i, \Delta j)^T \Big|_U. \quad (3)$$

In the initial state, a target is selected at the original point of the imaging plane; the target block will move when the MAV moves. In the searching area of the current frame, the optical flow vector of the target block can be obtained by calculating the minimum SAD between the current block and the previous block.

In the experiment, the images which are perpendicular to the camera are collected. In the whole collecting process, a data block which is  $8 \times 8$  pixel is used as the block matching object, and the searching area contains  $\pm 4$  pixels. So there

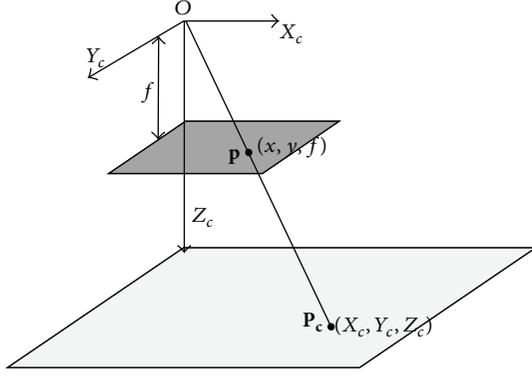


FIGURE 2: The model of pin-hole image plane.

are 64 pixel points and 81 candidate vector directions in each frame image. After obtaining each frame image, the mean absolute errors of each candidate vector are calculated and the minimum value is selected as the optical flow vector.

**2.2. The Model of Optical Flow.** The motion model of optical flow is projecting the three-dimensional motion onto the two-dimensional image plane of the camera. There are two common optical flow estimation models: one is pin-hole image plane approach which is derived from the principle of insect and vertebrate visual system; the other is spherical imaging surface approach which is derived from insect compound eyes. In our study, the pin-hole image plane approach is used to estimate the motion of MAV under geographic coordinate system.

The pin-hole image plane model is shown as in Figure 2.  $\mathbf{P}_c = [X_c, Y_c, Z_c]^T$  is set as a point under camera coordinate system;  $f$  stands for the focal length, so  $\mathbf{P}_c$  can be expressed as  $\mathbf{p} = [x, y, f]^T$ :

$$\mathbf{p} = f \frac{\mathbf{P}_c}{Z_c}, \quad (4)$$

$$x = f \frac{X_c}{Z_c}, \quad (5)$$

$$y = f \frac{Y_c}{Z_c}. \quad (6)$$

Considering any point  $\mathbf{P}$  on the ground, point  $\mathbf{P}$  has the following relationship relative to the MAV under the camera coordinate system:

$$\mathbf{V}_c = -\mathbf{T}_c - \boldsymbol{\omega} \times \mathbf{P}_c. \quad (7)$$

Equation (7) is expanded in three dimensions and (8) can be obtained:

$$\begin{aligned} V_{x_c} &= -T_{x_c} - (\omega_y Z_c - \omega_z Y_c), \\ V_{y_c} &= -T_{y_c} - (\omega_z X_c - \omega_x Z_c), \\ V_{z_c} &= -T_{z_c} - (\omega_x Y_c - \omega_y X_c), \end{aligned} \quad (8)$$

where  $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$  is angular velocity of MAV;  $\mathbf{T}_c = [T_{x_c}, T_{y_c}, T_{z_c}]^T$  is average velocity of MAV under camera coordinate system.

After derivation calculus to (4), the relationship between the velocity of  $\mathbf{P}_c$  under the camera coordinate system and the velocity of  $\mathbf{p}$  under the image plane can be obtained:

$$\frac{\text{flow}}{\Delta \text{time}} = \mathbf{v} = f \frac{Z_c \mathbf{V}_c - V_z \mathbf{P}_c}{Z_c^2}, \quad (9)$$

where  $\mathbf{v} = [v_x, v_y, v_z]^T$ , and (10)–(12) can be obtained by expanding (9):

$$v_x = \frac{f}{Z_c^2} (V_{x_c} Z_c - V_z Z_{x_c}), \quad (10)$$

$$v_y = \frac{f}{Z_c^2} (V_{y_c} Z_c - V_z Z_{y_c}), \quad (11)$$

$$v_z = 0. \quad (12)$$

Substituting (8) into (10) and (11), (13) can be obtained:

$$\begin{aligned} v_x &= \frac{T_{z_c} x - T_{x_c} f}{Z_c} - \omega_y f + \omega_z y + \frac{\omega_x x y - \omega_y x^2}{f}, \\ v_y &= \frac{T_{z_c} y - T_{y_c} f}{Z_c} + \omega_x f - \omega_x y + \frac{\omega_x y^2 - \omega_y x y}{f}. \end{aligned} \quad (13)$$

In (13),  $v_x$  and  $v_y$  are the optical flow components on directions  $x$  and  $y$ , which can be calculated by SAD and BMA.  $Z_c$  can be obtained by ultrasonic which was integrated in the optical flow sensor. Angular velocity  $\omega_x$ ,  $\omega_y$ , and  $\omega_z$  can be obtained by gyroscope;  $x$  and  $y$  can be substituted by (2) and (3). Therefore, the translational velocity  $\mathbf{T}_c$  of the MAV under the camera coordinate system can be estimated, and the MAV velocity under geographic coordinate system can be calculated through coordinate transformational matrix  $\mathbf{C}_c^n$ . After integration of velocity, the data for position of MAV can be obtained at last.

**2.3. Attitude Estimation Based on Accelerometer/Magnetometer.** By measuring the gravitational field, the accelerometer can determine the roll and pitch of the MAV under the condition of no own acceleration; by measuring the geomagnetic field, the magnetometer can determine the heading of MAV based on the vehicle's attitude information provided by accelerometer. Then, the whole attitude information without accumulated error can be obtained by integrating accelerometer/magnetometer.

**2.3.1. The Roll and Pitch Obtained by Accelerometer.** The component of gravity vector under the geographic coordinate system is  $[0 \ 0 \ -g]^T$ . When the vehicle is static (no relative acceleration to the navigation coordinate system), the measured value of accelerometer which is fixed in the carrier coordinate system is  $\mathbf{a}_b = [a_{x_b} \ a_{y_b} \ a_{z_b}]^T$ . The heading of the vehicle has no influence on the output of

accelerometers of directions  $x$  and  $y$  due to the gravitational acceleration perpendicular to horizontal plane. Therefore, it can be concluded that

$$\begin{bmatrix} a_{x_b} \\ a_{y_b} \\ a_{z_b} \end{bmatrix} = \begin{bmatrix} \cos \gamma & \sin \gamma \sin \theta & -\sin \gamma \cos \theta \\ 0 & \cos \theta & \sin \theta \\ \sin \gamma & -\cos \gamma \sin \theta & \cos \gamma \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}. \quad (14)$$

The roll and pitch can be calculated:

$$\begin{aligned} \theta &= \arcsin\left(-\frac{a_{y_b}}{g}\right), \\ \gamma &= \arctan\left(-\frac{a_{x_b}}{a_{z_b}}\right). \end{aligned} \quad (15)$$

This method uses the projection information of the earth's gravitational acceleration in carrier coordinate system to reflect the attitude information of the vehicle, so the above

$$\begin{bmatrix} H_{x_n} \\ H_{y_n} \\ H_{z_n} \end{bmatrix} = \begin{bmatrix} \cos \gamma \cos \varphi - \sin \gamma \sin \theta \sin \varphi & -\cos \theta \sin \varphi & \sin \gamma \cos \varphi + \cos \gamma \sin \theta \sin \varphi \\ \cos \gamma \sin \varphi + \sin \gamma \sin \theta \cos \varphi & \cos \theta \cos \varphi & \sin \gamma \sin \varphi - \cos \gamma \sin \theta \cos \varphi \\ -\sin \gamma \cos \theta & \sin \theta & \cos \gamma \cos \theta \end{bmatrix} \begin{bmatrix} H_{x_b} \\ H_{y_b} \\ H_{z_b} \end{bmatrix}, \quad (16)$$

where  $[H_{x_n} \ H_{y_n} \ H_{z_n}]^T$  can be referred to by [19]. In Taiyuan area (northern latitude  $37.8^\circ$ , east longitude  $112.5^\circ$ ),  $[H_{x_n} \ H_{y_n} \ H_{z_n}]^T$  is shown as (17), and the data of  $[H_{x_b} \ H_{y_b} \ H_{z_b}]^T$  can be obtained from the magnetometer:

$$\begin{aligned} H_{x_n} &= -0.216 \times 10^{-5}T, \\ H_{y_n} &= 0.309 \times 10^{-4}T, \\ H_{z_n} &= -0.43 \times 10^{-4}T. \end{aligned} \quad (17)$$

Supposing that the magnetic field intensity keeps on being constant during the flight of MAV, the heading of MAV under geographic coordinate system can be calculated by (16) and (17) based on the roll and pitch provided by accelerometer.

### 3. Data Fusion Based on EKF

**3.1. Optical Flow Sensor/Accelerometer Integrated System for Velocity and Position Measurements.** In our study, the EKF filter is employed to fuse the accelerometer and optical flow sensor. The velocity and position calculated by accelerometer under navigation coordinates are selected as the state value, and the velocity and position calculated by optical flow sensor are selected as the observer value. The estimation process is shown in Figure 3.

equations are correct only under the condition that there is no acceleration of the vehicle. Actually, the vehicles are not always of static or uniform motion, and the measured values of accelerometer are not equal to the component of gravitational acceleration under MAV coordinate system anymore once the carrier is under accelerated motion. Therefore, this method only can be used for measurement in static state, and it is necessary to find another attitude measuring method for dynamic conditions.

**2.3.2. The Heading Obtained by Magnetometer.** The component of earth's magnetic field intensity under geographic coordinate system is  $\mathbf{H}_n = [H_{x_n} \ H_{y_n} \ H_{z_n}]^T$ . The magnetometers are fixed on the carrier and the coordinate of the magnetometers is the same as the carrier coordinate system  $F_b$ , and the component of magnetic field intensity under the carrier coordinate system is  $\mathbf{H}_b = [H_{x_b} \ H_{y_b} \ H_{z_b}]^T$ . The projection of magnetic field intensity to the geographic and camera coordinate systems can be presented as

Considering the nonlinear system state equation and observed equation,

$$\begin{aligned} \mathbf{X}_{k+1} &= f[\mathbf{X}_k, k] + \mathbf{B}_k \mathbf{U}_k + \mathbf{\Gamma}_k \mathbf{W}_k, \\ \mathbf{Z}_{k+1} &= h[\mathbf{X}_{k+1}, k+1] + \mathbf{V}_{k+1}, \end{aligned} \quad (18)$$

where  $\mathbf{X}_k = [X_n \ Y_n \ Z_n \ V_{x_n} \ V_{y_n} \ V_{z_n}]^T$  is state vector which includes the velocity information and position information of MAV;  $\mathbf{Z}_k = [x \ y \ Z_c]^T$  is observer vector which includes the optical flow data of directions  $x$  and  $y$  from optical flow sensor and the data  $Z_c$  obtained from ultrasonic sensors.  $\mathbf{U}_k = [0 \ 0 \ 0 \ a_{x_n} \ a_{y_n} \ a_{z_n}]^T$  is the control vector of system that can be obtained by the accelerometer data after coordinate matrix transformation;  $\mathbf{B}_k$  is matrix of control allocation;  $\mathbf{\Gamma}_k$  is matrix of noise allocation;  $\mathbf{W}_k$  is matrix of process noise;  $\mathbf{V}_k$  is observer noise;  $f$  represents the system state function;  $h$  represents the observation function.

Substituting the state and observed equation into EKF, the time renewal equation is

$$\begin{aligned} \widehat{\mathbf{X}}_{k+1/k} &= f[\mathbf{X}_k, k] + \mathbf{B}_k \mathbf{U}_k, \\ \mathbf{P}_{k+1/k} &= \mathbf{\Phi}_{k+1,k} \mathbf{P}_k \mathbf{\Phi}_{k+1,k}^T + \mathbf{\Gamma}_k \mathbf{Q}_k \mathbf{\Gamma}_k^T. \end{aligned} \quad (19)$$

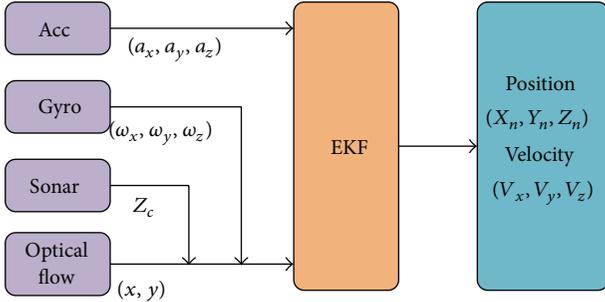


FIGURE 3: Optical flow sensor/accelerometer integrated system.

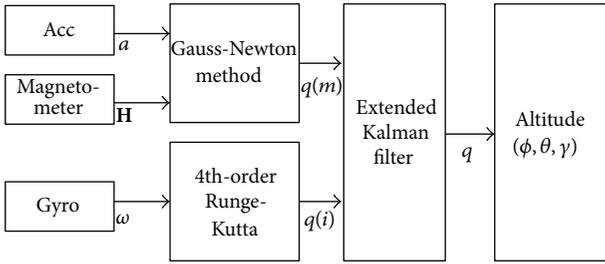


FIGURE 4: Gyroscope/accelerometer/magnetometer integrated system.

Measurement renewal equation is

$$\begin{aligned} \mathbf{K}_{k+1} &= \mathbf{P}_{k+1/k} \mathbf{H}_{k+1} (\mathbf{H}_{k+1} \mathbf{P}_{k+1/k} \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1})^{-1}, \\ \hat{\mathbf{X}}_{k+1} &= \hat{\mathbf{X}}_{k+1/k} + \mathbf{K}_{k+1} \{ \mathbf{Z}_{k+1} - h[\hat{\mathbf{X}}_{k+1/k}, k+1] \}, \\ \mathbf{P}_{k+1} &= (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \mathbf{P}_{k+1/k} (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1})^{-1} \\ &\quad + \mathbf{K}_{k+1} \mathbf{R}_{k+1} \mathbf{K}_{k+1}^T, \end{aligned} \quad (20)$$

where  $\Phi_{k+1,k} = (\partial f[\mathbf{X}_k, k] / \partial \mathbf{X}_k^T)|_{\mathbf{X}_k = \hat{\mathbf{X}}_k}$  and  $\mathbf{H}_{k+1,k} = (\partial h[\mathbf{X}_{k+1}, k+1] / \partial \mathbf{X}_{k+1}^T)|_{\mathbf{X}_{k+1} = \hat{\mathbf{X}}_{k+1/k}}$ .

From the process of EKF mentioned above, the data for position and velocity of MAV can be obtained in geographic coordinate system.

**3.2. Gyroscope/Accelerometer/Magnetometer Integrated System Based on EKF.** The attitude of MAV can be obtained through the integration of the angular rate from gyroscope output signal, however, the performance of MEMS gyroscope would be influenced by drift. The integrated accelerometer/magnetometer system can provide attitude information without drift; therefore, it is necessary to fuse the data of multisensors by using EKF. The filtering process is shown as Figure 4; the system state vector can be expressed as  $\mathbf{X}_k = [q_0 \ q_1 \ q_2 \ q_3 \ \omega_x \ \omega_y \ \omega_z]^T$ , where  $[q_0 \ q_1 \ q_2 \ q_3]^T$  is quaternion of the system state; it can be calculated by fourth-order Runge-Kutta method.  $[\omega_x \ \omega_y \ \omega_z]^T$  is the output of gyroscope.

The system observation vector can be expressed as  $\mathbf{Z}_k = [q_0 \ q_1 \ q_2 \ q_3]^T$ , which can be calculated by Gauss-Newton method based on the data of accelerometer and magnetometer.

**3.3. Optical Flow Sensor/INS/Magnetometer Integrated Navigation Scheme.** The effective estimation of MAV velocity and position can be obtained by using optical flow sensor/INS integrated navigation system based on EKF no matter under stationary status or motion status. According to the characteristics of attitude measurements by using gyroscope and accelerometer/magnetometer, accelerometer and magnetometer are integrated to calibrate the gyroscope during stationary or uniform motion state; when the motion state is detected as accelerating or decelerating, the standalone gyroscopes are used to obtain the attitude by strap-down calculating. The motion state is detected by optic flow sensor. The overall navigation scheme is shown in Figure 5.

## 4. Experiments and Verification

In order to verify the proposed optical flow sensor/INS/magnetometer integrated navigation system, a MAV flight experiment was carried out. The experiment location was in the stadium of North University of China, Taiyuan. As shown in Figure 6, we can see that, during the flight procedure, the MAV kept flying to the north. The flying distance was 50 meters, and the flight height is 1.5 meters.

**4.1. Experiments Platform.** As shown in Figures 7 and 8, the autonomous MAV is used as flight platform (Figure 7(a)); STM32F103Z is selected as the flight control processor (Figure 7(b)); the optical flow sensor is produced by 3D Robotics Corporation which is named as PX4FLOW (Figure 7(c)); the IMU is MPU6050 (Figure 7(d)); the magnetometer is HMC3883L which is produced by Honeywell Limited (Figure 7(d)); DJI 2312 which is produced by DJI is selected as the MAV motor (Figure 7(e)); and HC-12 wireless serial interface module (Figure 7(f)) is used to transmit the sensor's data from MAV to PC.

**4.2. Verification Results.** Figures 9–11 are the velocity estimation results, position estimation results, and attitude estimation results, respectively. Table 1 is the end position comparison results between INS only and OFS/INS/magnetometer. It can be concluded that (1) in the process of velocity estimation and position estimation, the OFS/INS/magnetometer based on EKF can decrease the drift of INS effectively. For example, from Figure 9 we can see that the position drift is about 8 meters after 50 meters trajectory without the correction of OFS and magnetometer, and the position drift is decreased to less than 1 meter after using the integrated algorithm and (2) in the process of attitude estimation, the roll angle, pitch angle, and the yaw angle can be effectively tracked by OFS/INS/magnetometer based on EKF during static and uniform velocity status. And after nonuniform velocity motion, the attitude can be corrected immediately.

TABLE 1: Comparison results between INS only and OFS/INS/magnetometer at the end position.

Directions	Ground truth position at the end		INS only		OFS/INS/magnetometer	
	Ground truth value	Measurements	Error	Estimation	Error	
X	0	-1.87	—	0.25	—	
Y	50	57.8	15.6%	50.9	1.8%	
Z	0	0.78	—	0.14	—	

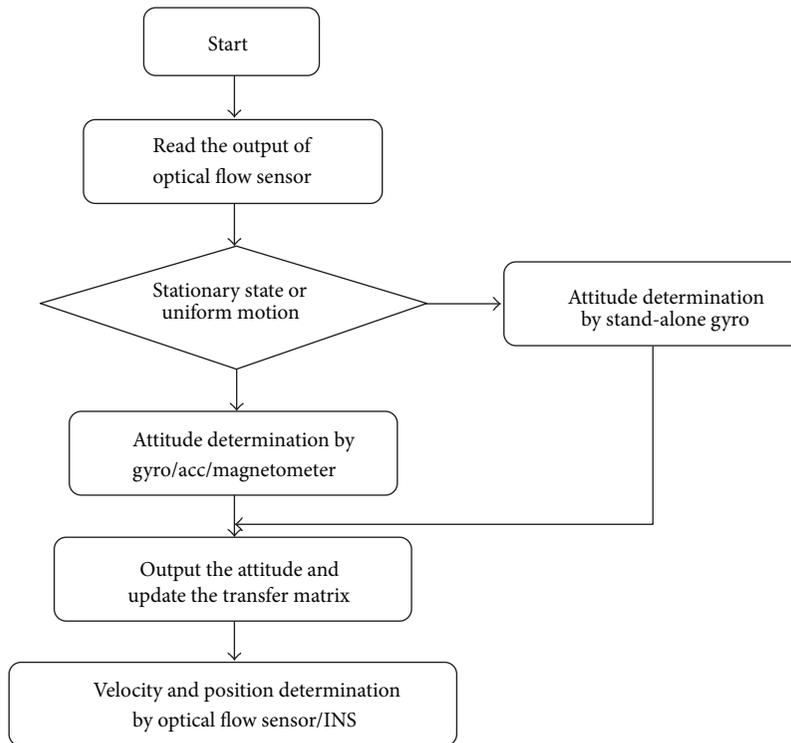


FIGURE 5: The flow chart of optical flow sensor/INS/magnetometer integrated navigation system.



FIGURE 6: The experiment trajectory.

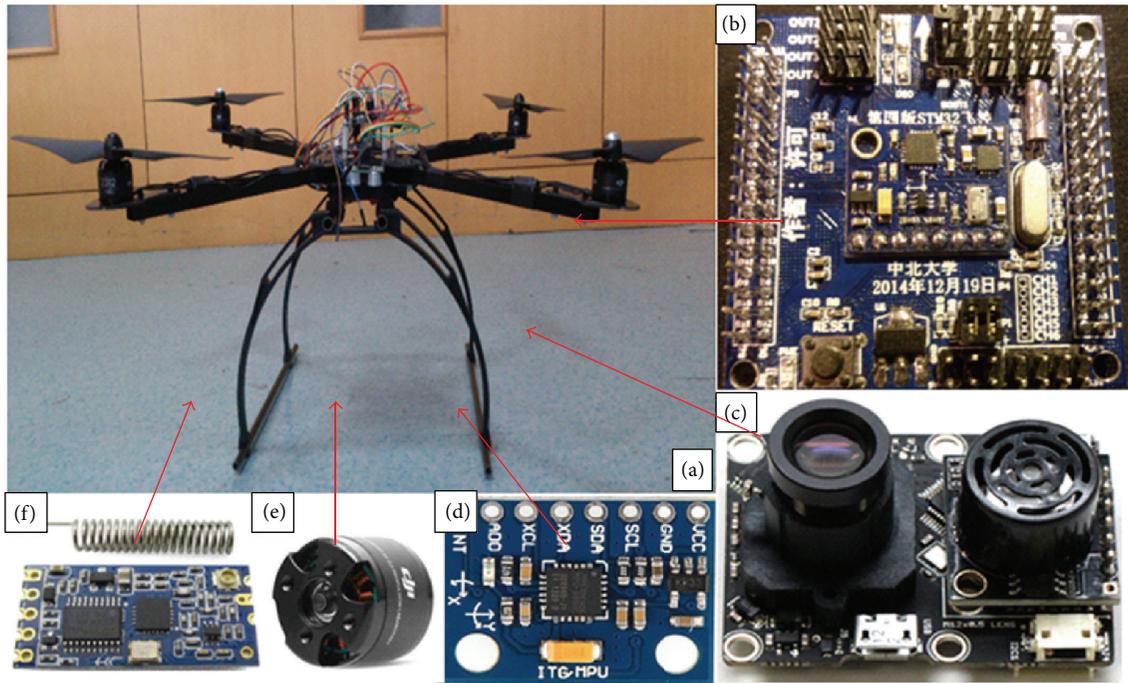


FIGURE 7: The hardware experiment platform.

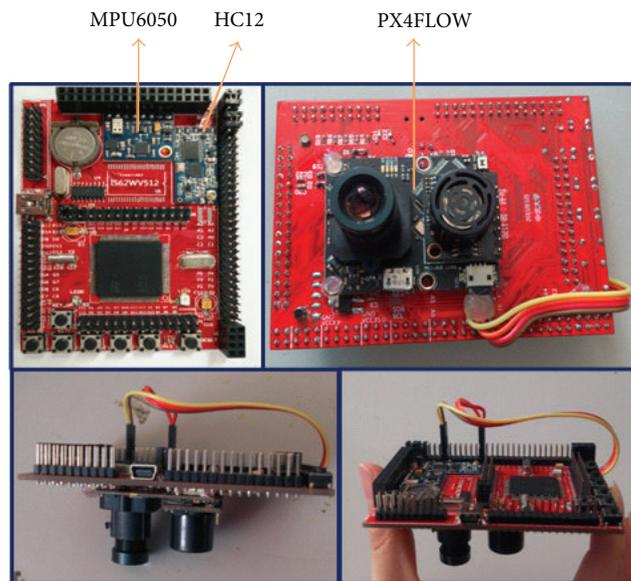


FIGURE 8: The INS/OFS/magnetometer integrated navigation system.

Figure 12 is the trajectory comparison results between ground truth, INS only, and EKF. It can be concluded that the proposed optical flow sensor/INS/magnetometer integrated navigation system can provide a higher accurate navigation result compared to INS only navigation system.

### 5. Conclusion

In this paper, an optical flow sensor/INS/magnetometer integrated navigation system is proposed for MAV. The

proposed method, which is based on EKF, combines optical flow sensor, INS, and magnetometer information to estimate the attitude, velocity, and position of MAV. Specifically, the gyro, accelerator, and magnetometer information are fused together to estimate the MAV attitude when the MAV is static or uniformly moving; and the gyro only is used to estimate the MAV attitude when the MAV is accelerating or decelerating. Experiment results show that the proposed method which can significantly reduce the errors for navigation position, velocity, and attitude, compared with the INS only navigation

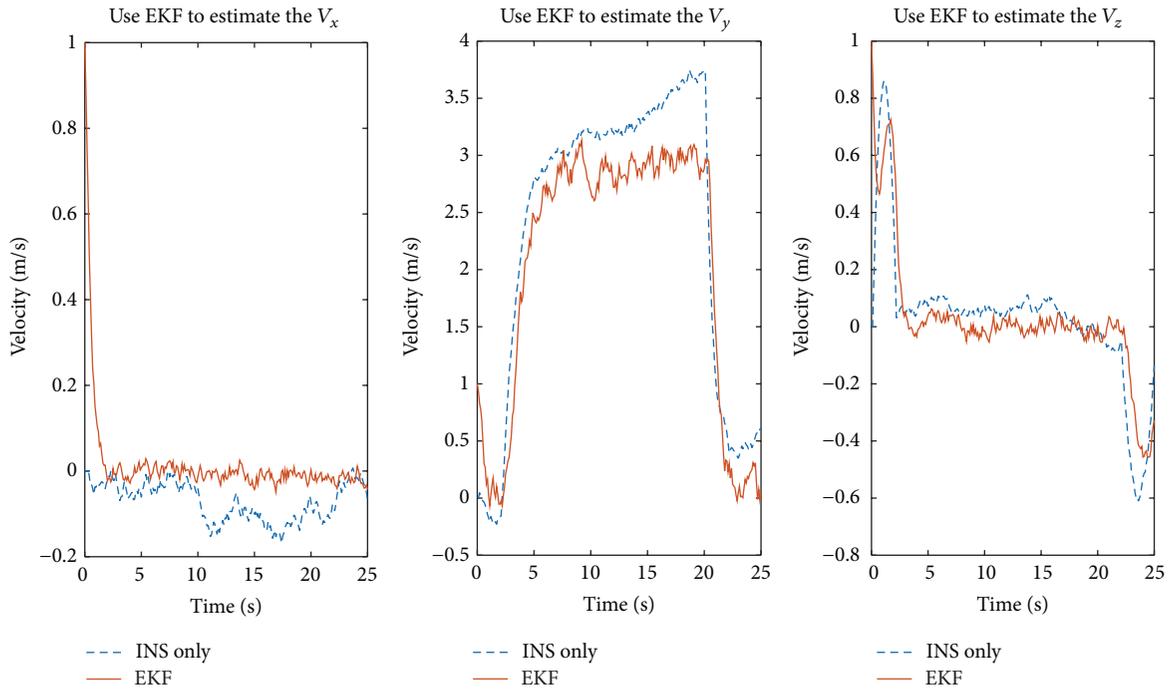


FIGURE 9: The velocity estimation results.

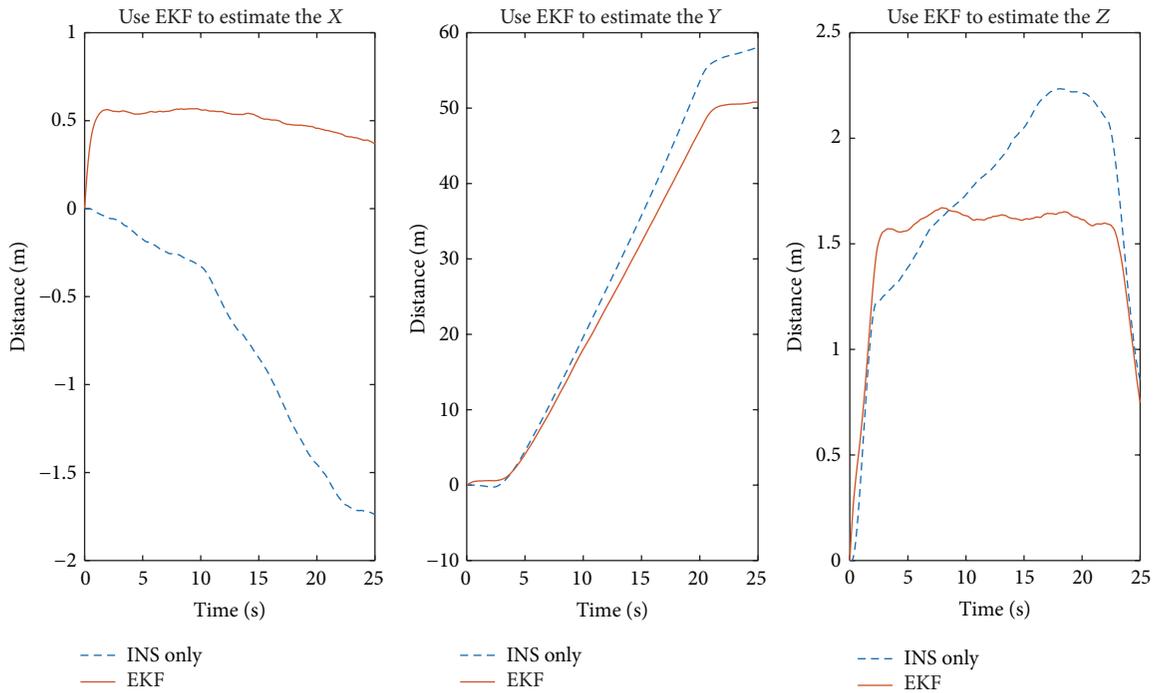


FIGURE 10: The position estimation results.

system, can effectively improve the navigation performance of MAV with a significant value of engineering application.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgments

This work was supported in part by the China National Funds for Distinguished Young Scientists (51225504), the National 973 Program (2012CB723404), the National Natural Science Foundation of China (91123016, 61171056), the Research Project Supported by Shanxi Scholarship Council of China

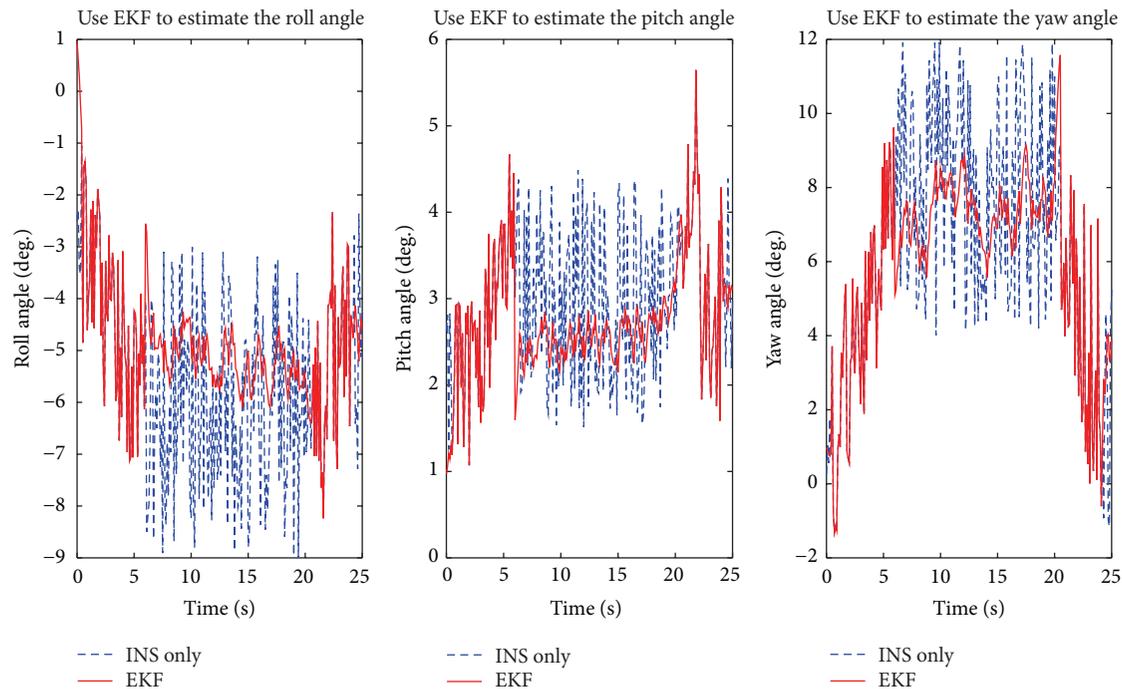


FIGURE 11: The attitude estimation results.

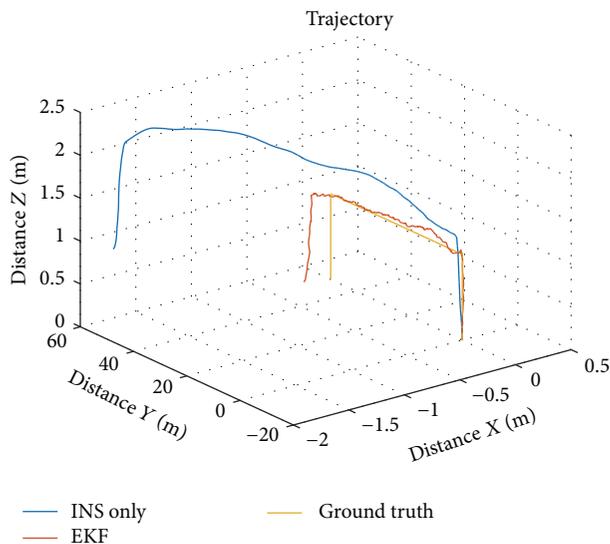


FIGURE 12: Trajectory comparison results between ground truth, INS only, and EKF.

(2015-082), and the College Funding of North University of China (110246).

## References

- [1] C. Friedman, I. Chopra, and O. Rand, "Indoor/outdoor scan-matching based mapping technique with a helicopter MAV in GPS-denied environment," *International Journal of Micro Air Vehicles*, vol. 7, no. 1, pp. 55–70, 2015.
- [2] Y. Hu, H. L. Zhang, and C. Tan, "The effect of the aerofoil thickness on the performance of the MAV scale cycloidal rotor," *Aeronautical Journal*, vol. 119, pp. 343–364, 2015.
- [3] M. Samani, M. Tafreshi, I. Shafieenejad, and A. A. Nikkhah, "Minimum-time open-loop and closed-loop optimal guidance with GA-PSO and neural fuzzy for Samarai MAV flight," *IEEE Aerospace and Electronic System Magazine*, vol. 30, pp. 28–37, 2015.
- [4] G. Flores, S. T. Zhou, R. Lozano, and P. Castillo, "A vision and GPS-based real-time trajectory planning for a MAV in unknown and low-sunlight environments," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 59–67, 2014.
- [5] A. Ali and N. El-Sheimy, "Low-cost MEMS-based pedestrian navigation technique for GPS-denied areas," *Journal of Sensors*, vol. 2013, Article ID 197090, 10 pages, 2013.
- [6] X. Chen, C. Shen, W.-B. Zhang, M. Tomizuka, Y. Xu, and K. Chiu, "Novel hybrid of strong tracking Kalman filter and wavelet neural network for GPS/INS during GPS outages," *Measurement*, vol. 46, no. 10, pp. 3847–3854, 2013.
- [7] T. Zhang and X. S. Xu, "A new method of seamless land navigation for GPS/INS integrated system," *Measurement*, vol. 45, no. 4, pp. 691–701, 2012.
- [8] R. Song, X. Y. Chen, C. Shen, and H. Zhang, "Modeling FOG drift using back-propagation neural network optimized by artificial fish swarm algorithm," *Journal of Sensors*, vol. 2014, Article ID 273043, 6 pages, 2014.
- [9] H. Q. Huang, X. Y. Chen, Z. K. Zhou, Y. Xu, and C. P. Lv, "Study of the algorithm of backtracking decoupling and adaptive extended kalman filter based on the quaternion expanded to the state variable for underwater glider navigation," *Sensors*, vol. 14, no. 12, pp. 23041–23066, 2014.
- [10] D. H. Yi, T. H. Lee, and D. I. Cho, "Afocal optical flow sensor for reducing vertical height sensitivity in indoor robot localization and navigation," *Sensors*, vol. 15, no. 5, pp. 11208–11221, 2015.

- [11] H. Chao, Y. Gu, and M. Napolitano, "A survey of optical flow techniques for UAV navigation applications," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS '13)*, pp. 710–716, IEEE, Atlanta, Ga, USA, May 2013.
- [12] F. Martínez, A. Manzanera, and E. Romero, "Automatic analysis and characterization of the hummingbird wings motion using dense optical flow features," *Bioinspiration & Biomimetics*, vol. 10, no. 1, Article ID 016006, 2015.
- [13] M. P. Rodriguez and A. Nygren, "Motion estimation in cardiac fluorescence imaging with scale-space landmarks and optical flow: a comparative study," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 2, pp. 774–782, 2015.
- [14] W. D. Ding, J. L. Wang, S. L. Han et al., "Adding optical flow into the GPS/INS integration for UAV navigation," in *Proceedings of the International Global Navigation Satellite Systems Society*, pp. 1–13, Surfes Paradise, Australia, 2009.
- [15] D. A. Mercado, G. Flores, P. Castillo, J. Escareno, and R. Lozano, "GPS/INS/Optic flow data fusion for position and velocity estimation," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS '13)*, pp. 486–491, Atlanta, Ga, USA, May 2013.
- [16] M. B. Rhudy, H. Y. Chao, and Y. Gu, "Wide-field optical flow aided inertial navigation for unmanned aerial vehicles," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '14)*, pp. 674–679, IEEE, Chicago, Ill, USA, September 2014.
- [17] K. Gageik, M. Strohmeier, and S. Montenegro, "An autonomous UAV with an optical flow sensor for positioning and navigation," *International Journal of Advanced Robotic Systems*, vol. 10, pp. 1–9, 2013.
- [18] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, "An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '13)*, pp. 1736–1741, Karlsruhe, Germany, May 2013.
- [19] H.-D. Zhao, H.-S. Cao, J.-Z. Zhu, and H.-S. Zhang, "Method of attitude measurement with magnetometer and gyroscope," *Journal of North University of China*, vol. 31, no. 6, pp. 631–635, 2010 (Chinese).

## Research Article

# Intraframe Scene Capturing and Speed Measurement Based on Superimposed Image: New Sensor Concept for Vehicle Speed Measurement

Mate Nemeth<sup>1,2</sup> and Akos Zarandy<sup>1,2</sup>

<sup>1</sup>Computational Optical Sensing and Processing Laboratory, Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA-SZTAKI), Budapest 1111, Hungary

<sup>2</sup>Faculty of Information Technology, Pazmany Peter Catholic University (PPKE-ITK), Budapest 1083, Hungary

Correspondence should be addressed to Mate Nemeth; [nemeth.mate@sztaki.mta.hu](mailto:nemeth.mate@sztaki.mta.hu)

Received 22 October 2015; Accepted 30 November 2015

Academic Editor: Fadi Dornaika

Copyright © 2016 M. Nemeth and A. Zarandy. 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 vision based vehicle speed measurement method is presented in this paper. The proposed intraframe method calculates speed estimates based on a single frame of a single camera. With a special double exposure, a superimposed image can be obtained, where motion blur appears significantly only in the bright regions of the otherwise sharp image. This motion blur contains information of the movement of bright objects during the exposure. Most papers in the field of motion blur are aiming at the removal of this image degradation effect. In this work, we utilize it for a novel speed measurement approach. An applicable sensor structure and exposure-control system are also shown, as well as the applied image processing methods and experimental results.

## 1. Introduction

Nowadays, an increasing tendency can be noticed for automation and integration of information and communication technologies into conventional services and solutions, in nearly every aspect of our lives. Car industry is one of the leading sectors of this evolution with the intelligent vehicle concept, as there are several already existing solutions for the assistance of the vehicle's operator (e.g., parking assist systems). Improved sensing technologies could also be used in the smart cities of the future, to improve traffic management and to provide real-time information to each individual vehicle, for better traffic load balancing. An imager sensor utilizing the proposed, novel speed measurement concept could be used as a sensing node of a distributed sensor network, as it is based on a low-cost sensor module.

Conventional speed measurement systems are usually based on either RADAR or LIDAR speed guns [1]. Both techniques use active sensing technologies, which are more complicated and expensive than passive camera systems. On

the other hand, there are methods in the literature aiming at producing reliable speed estimates, based on optical information only [2–5]. Scientific studies in this field can be divided into two major research directions: optical flow (interframe) and motion-blur (intraframe) based displacement calculation methods; however, there are only a few papers related with the latter case [3]. Besides speed measurement, it would be profitable for many possible applications to be capable of identifying cars by number plate recognition. Therefore, it is an essential feature of these systems, to provide adequate image quality. The most important drawback of the motion-blur based methods is that the measurement concept itself is based on the degradation of the image, which is controversial with precise number plate identification, although in [3] a deblurring method is presented capable of providing appropriate image quality, with a sensor utilizing fast shutter speed, and high resolution. Our approach is based on a completely different measurement principle, using a low-end imager sensor. In this paper, we propose a novel double-exposure method, based on a special imager chip for intraframe speed

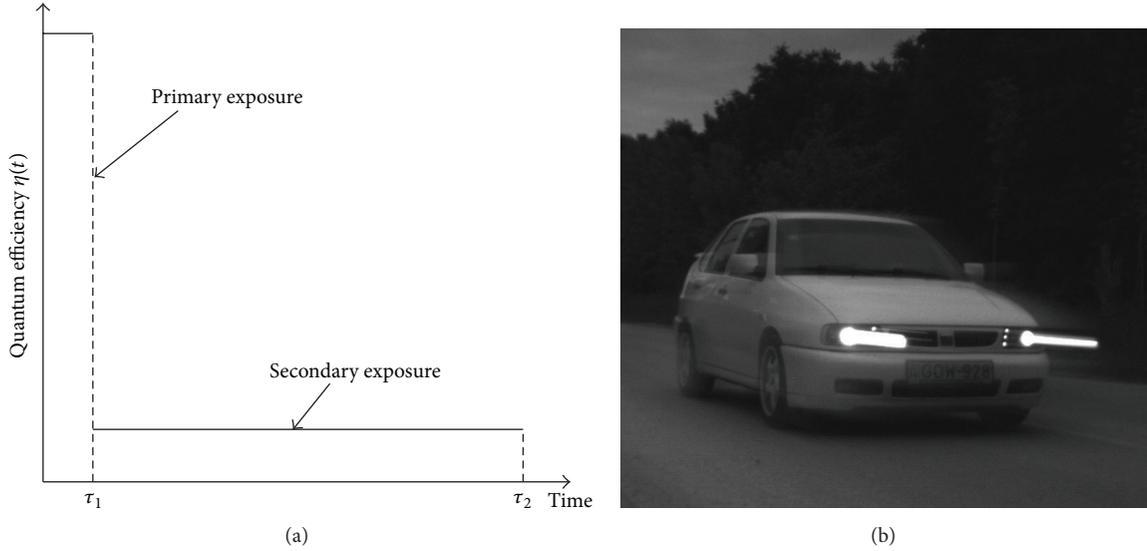


FIGURE 1: (a) Exposure-control scheme of the proposed method, with the primary  $[0, \tau_1]$  and secondary  $[\tau_1, \tau_2]$  exposure intervals. (b) Expected superimposed image with the applied exposure-control scheme. The bright headlights generate the saturated traces.

measurement, which meets the mentioned requirements. A suitable sensor structure is shown along with hardware-level control for the imager.

The paper is composed in the following way. In Section 2, the fundamental concept is described, and the speed estimation based on the displacement is formulated. Section 3 presents a suitable pixel-level control method for the measurements and requirements related to the image itself. The applied image processing algorithms and compensation methods are described in Sections 4 and 5, and the paper is summarized with a conclusion.

## 2. Concept Formulation

The amount of incident light reaching the imager sensor is determined by the camera's shutter speed ( $t$ ), the lens relative aperture ( $N$ ), and the luminance of the scene ( $L_v$ ). Considering a measurement situation where  $N, L_v$  are given, the intraframe behavior of fast moving objects on the image plane can be controlled through shutter speed. The appearing motion blur on an image is proportional to the speed of the object and the shutter speed.

**2.1. Measurement Concept.** Our speed measurement concept is based on a special control method of the sensor shutter. The proposed method ensures adequate image quality, while still holding information describing intraframe motion of certain objects with very bright spots. The classical shutter cycle of the CMOS sensor (open, close) is expanded with an intermediate, semiopen state. We defined a double-exposure scheme (Figure 1), with each phase having different quantum efficiency (QE) values. Quantum efficiency  $\eta$  describes the responsivity of an image sensor, and it is defined as the

number of photogenerated carriers per incident photon [6, 7] as described in

$$\eta = \frac{J_{ph}/q}{P_{ph}/\hbar\omega}, \quad (1)$$

where  $J_{ph}$  is the light-induced current density and  $P_{ph}$  is the optical power per unit area of the incident light. The QE of a specific sensor with respect to wavelength can be found in its datasheet.

The first phase of the double exposure is denoted with  $\tau_1$ . This is a short interval, when the electronic shutter is fully open. During this time, the dominant component of the integrated image is collected. Since  $\tau_1$  is small, even the moving objects will not be blurred. Then, in the semiopen phase  $[\tau_1, \tau_2]$ , the process continues with significantly longer exposure ( $\tau_1 \ll \tau_2$ ), but with a lower QE. This means that much less portion of the incident light will generate charge carriers in the photodiode in a time unit, reducing the responsivity of the sensor. Assuming that we can control the length of the double-exposure phases ( $\tau_1$  and  $\tau_2$ ), we can generate a superimposed image, consisting of a sharp image, and a blurred image. On the blurred image, only the high intensity regions of the scene appear, which typically drive the pixels to saturation or to a near saturation value. In the case of a fast moving object with a light source (e.g., car on a highway with headlights on), this implies that a light trace appears on the image plane (Figure 1) according to the movement path of the light source during exposure, and the length of the trace is proportional to the speed of the object.

**2.2. Calculation of Speed Estimates.** The measurement geometry is presented in Figure 2. Considering  $\gamma, e, c$  are known,

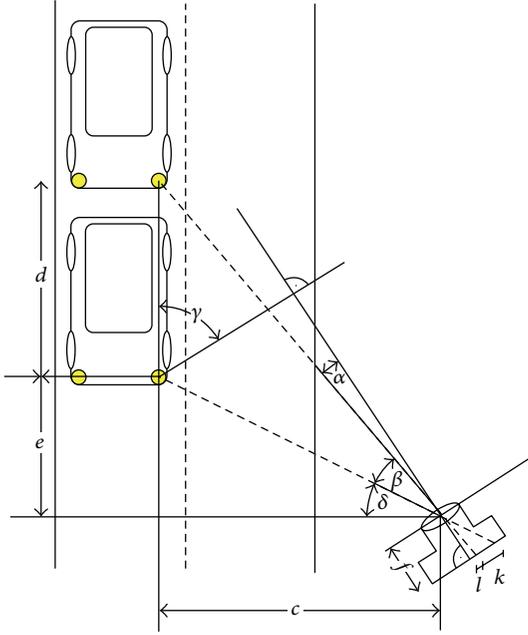


FIGURE 2: Geometry of the measurement setup.

as spatial geometry is known prior to the measurement, one can derive (2) and (3) from the given geometry:

$$\frac{e}{c} = \tan \delta, \quad (2)$$

$$\frac{e+d}{c} = \tan(\beta + \delta), \quad (3)$$

$$\delta = \gamma - \alpha - \beta, \quad (4)$$

where  $\gamma$  is the angle between the image plane and the movement direction of the measured object,  $d$  is the displacement, and  $\alpha, \beta$  can be derived from the image, assuming that the calibration parameters of the camera are known. After substitution of (2) and (4) into (3) and removing  $e$  and  $\delta$ , we have

$$d = c(\tan(\gamma - \alpha) - \tan(\gamma - \alpha - \beta)). \quad (5)$$

If the interval of the secondary exposure (or intraframe time) is denoted as  $[\tau_1, \tau_2]$ , movement speed of the measured object can be obtained as follows:

$$v = \frac{d}{\tau_2 - \tau_1}. \quad (6)$$

As a result, the expected accuracy of the speed measurement is proportional to the measurement accuracy of the light trace on the image plane (again, if we consider that spatial geometry and camera parameters are known). Hence the longer the light trace is, the more accurately its length can be measured. The lateral movement of the measured vehicle inside the lane was considered to be neglectable.

### 3. Implementation

CMOS sensor technology enables the implementation of various pixel-level control or computation circuits. Therefore, special electronic shutters can be implemented with pixel-level exposure-control circuitry. This section presents a novel exposure-control concept for CMOS sensors, to implement the described double-exposure method.

Most CMOS imagers apply rolling shutter [6], where the exposure starts in a slightly delayed manner for every row of the sensor. This causes geometrically incoherent images when capturing moving objects. Therefore, in some machine vision applications, rolling shutter cameras are not applicable. This fact calls for the other type of CMOS sensors featuring global shutter pixels (Figure 3). In this case, the integration of the pixels in the entire array is performed simultaneously, and the readout is performed in a row-by-row manner.

**3.1. Description of a Global Shutter Pixel.** A fundamental component of a global shutter (or snapshot) pixel [6] is a sample-and-hold (S/H) switch with analog storage (all parasitic capacitances in the amplifier input) and a source follower amplifier, which acts as a buffer amplifier to isolate the sensing node, and performs the in-pixel amplification. The row select (RS) transistor plays an important role in the readout phase of the exposure cycle. The schematic figure of a common 4T global shutter pixel (pixel realization using 4 transistors) is shown in Figure 3. The incident light generated charge is transferred from the PD and stored in the in-pixel parasitic capacitance after the integration.

**3.2. Pixel Control.** To ensure that the sensor operates in accordance with the double-exposure schedule, the S/H stage could be replaced with a suitable control circuitry, which implements the functionality of the semiopen state of the shutter.

One important issue related to charge storage is the global shutter efficiency (GSE). According to [8–10], an increasing tendency of CMOS imager manufacturers can be noticed to achieve better GSE values, which is defined as a ratio of photodiode sensitivity during open state to pixel storage parasitic sensitivity during closed state. Or in other words, it is the ratio of the QE in the open state to the QE in the close state of the shutter. The storage parasitic sensitivity has many components, including charge formed in the storage due to direct photons, diffusion charge parasitic current outside of photodiode (PD), and direct PD to analog storage leakage. The GSE of a specific CMOS sensor [8] (Aptina MT9M021) used in this study is shown in Figure 3. Maintaining sensor performance, while reducing the pixel size, requires higher quantum efficiency and lower noise floor. Electrical and optical isolation of the in-pixel storage nodes is also becoming more and more difficult with the shrinking pixel size. Aptina recent 3.75 and 2.8  $\mu\text{m}$  (3rd and 4th generation) global shutter pixel arrays implement [8] some extra features like Row-wise Noise Correction and Correlated Double Sampling (CDS) to reduce the impact of dark current (thermal generation of electron-hole pairs in the depleted region) and readout noise and to improve GSE. On the other hand increasing pixel-level

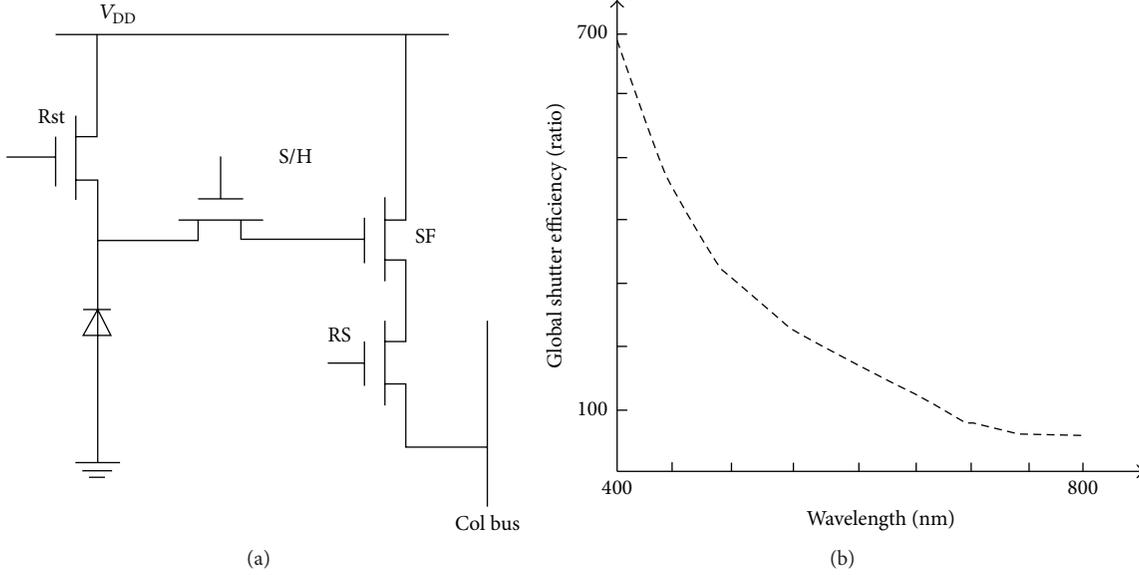


FIGURE 3: (a) Transistor scheme of a common global shutter pixel; S/H: sample-and-hold; SF: source follower; RS: row select. (b) Global shutter efficiency of an Aptina 3rd generation  $3.75 \mu\text{m}$  pixel [8].

functionality, along with transistor number, is controversial with sensitivity, since the fill factor is decreasing.

In our experiments, we exploit the relatively low GSE of the Aptina MT9M021 sensor. At short integration times and low scene luminance, PD to analog storage leakage during the readout phase could emulate the low QE phase of the double-exposure method proposed in Section 2.1 (assuming that  $\tau_1$  and  $\tau_2$  are represented by the exposure time and the read-out time, resp.). In our experiments, we used a custom test hardware (described in Section 3.3), where we can control not only the integration time of the sensor but the readout time (through readout frequency) as well. Qualitative characteristics of the secondary blurred image which will be superimposed on the initial sharp image depend on the readout time ( $T_{\text{readout}}$ ). Read-out time can be calculated as follows:

$$T_{\text{readout}} = \frac{1}{f_{\text{pixclk}}} \times N_{\text{row}} \times \text{Row}_{\text{length}}, \quad (7)$$

where  $f_{\text{pixclk}}$  denotes the readout frequency. As a result, (6) can be rewritten into the following form:

$$v = \frac{d}{T_{\text{readout}}}. \quad (8)$$

Notice that (7) implies that the readout time of a detected object depends on its vertical position on the image.

The capabilities of our hardware enables us to specify the intervals of  $[0, \tau_1]$  and  $[\tau_1, \tau_2]$ , based on the QE and the GSE of the specific sensor. During the measurements, we made an empirical observation. The trade-off between the license plate readability and the contrast of the background and the light trace is balanced, when the following statement holds:

$$\int_0^{\tau_1} \eta(t) dt \approx \int_{\tau_1}^{\tau_2} \eta(t) dt. \quad (9)$$

This needs further investigation, but, in this case, the stored charge of the primary exposure and charge accumulation caused by the leakage (until readout) is in the same order of magnitude (9). As a result, a bright trace will appear on the image, which represents the movement of the headlight during the readout. Technical details connected with the imager setup are described in Section 4.1.

**3.3. Test Hardware.** In our experiments, we used a custom test hardware, described in detail in [11, 12]. Figure 4 shows the camera module and the image capturing device. The system consists of a camera module, an interface card, and an FPGA development board. The camera module utilizes the previously mentioned Aptina MT9M021 sensor, which is operated in trigger mode, so that multiple cameras can be synchronized at hardware level. The interface card is responsible for deserializing the camera data and providing the FPGA board with input. This interface card is designed to be compatible with a series of FPGA development boards. In our experiment, the used FPGA board was Xilinx's SP605 Evaluation Kit based on a Spartan-6 FPGA. As stated in Section 3.2, we can control the readout frequency of the sensor, which makes it an ideal platform for the measurements.

## 4. Light Trace Detection

Let us consider the measurement geometry ( $\gamma, e, c$  on Figure 2) to be known. As described in Section 2.1, the expected accuracy of the method inherently depends on the accuracy of the light trace length measurement on the captured images. Hence, a crucial point of the whole system is the precise trace detection method. To specify the requirements of such system, the related regulations and specificities of the possible applications has to be taken into consideration.

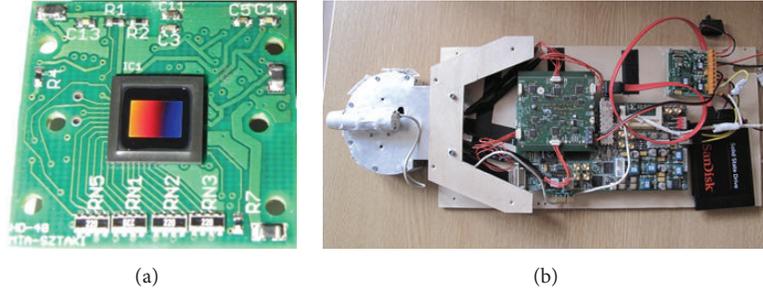


FIGURE 4: (a) Designed camera module utilizing the Aptina MT9M021 CMOS imager sensor. (b) Image capturing system, the cameras are placed inside an aluminum holder frame, connected to the FPGA board.

The first obvious application could be to use the system as a speed cam. The regulations in this regard vary with different countries; for example, in the United States, the Unit Under Test must display the speed of a target vehicle within  $\pm 2$ ,  $-3$  km/h, according to US Department of Transportation National Highway Traffic Safety Administration [13]. We will use this data as a reference benchmark during the research, just for initial proof of concept measurements, without any approved validation process. Notice that the specification is more tolerant at lower speed ranges in terms of relative accuracy. Actually this absolute precision requirement is matching our speed measurement concept, because  $\pm \epsilon$  pixel accuracy of the light trace detection is equivalent to  $\pm v_\epsilon$  tolerance.

Besides speed cams there can be other applications with less strict requirements, especially in a smart city environment, like in the field of traffic statistics and traffic monitoring.

**4.1. Input Image Requirements and Description of the Gathered Database.** To achieve the best results in the light trace detection process, the input image has to be captured with the appropriate imager sensor settings. The integration time and readout time of the sensor fundamentally changes the effectiveness of the trace detection method. The trace measurement method would require as short integration time as possible. This would ensure the maximum contrast between the light trace and the background, making the detection much more easier and accurate. On the contrary, image segmentation for license plate recognition needs a brighter image, with longer integration time. On the other hand, it would be profitable to prolong the readout time, because the longer the trace is, the more accurate the measurement becomes. But as the secondary exposure becomes longer, more charge is accumulated, blurring the image in the lower intensity regions also (even with lower QE), making the car identification more difficult. In our experiments, we observed the best results at a relatively low illumination range of 100–1700 lx, and all of the images presented in the paper were captured in these lighting conditions. If the illumination exceeds this level and the lower limit of the integration time of the sensor does not allow further compensation, a neutral density filter should be used to maintain the quality of the results. During our measurements, we used integration

times around 0.2 ms with 22 MHz readout frequency, which applies to the previously mentioned illumination level and satisfies assumption (9) in the following way. Consider that the measured object is in the middle of the frame. After rewriting (9), we get

$$\eta \times T_{\text{integration}} \approx \frac{\eta}{\text{GSE}} \times T_{\text{readout}}, \quad (10)$$

where GSE is an average efficiency value in the visible spectrum. Combining (10) with (7) and after substitution with the specific values of our sensor we get (11)

$$0.2 \times 10^{-3} \times \eta \approx \frac{1}{22 \times 10^6} \times 500 \times 1650 \times \eta \times \frac{1}{200}. \quad (11)$$

We captured image sets for the image processing methods in a real measurement scenario. After selecting a suitable location for the measurement, we observed the passing traffic. Numerous images were captured with our test platform with a wide variety of vehicle and headlight combinations: passenger cars and vans with LED and halogen lamps. A collection of such images can be seen in Figure 5. The speed of the vehicles was around 40–60 km/h, since the measurements were performed in an urban area. Two separate image databases have been captured: a single camera and a stereo set, consisting of about 200 and 50 images, respectively. The single camera database has been separated to an evaluation set and a learning set. The learning image set consists of about 30 images of vehicles with different headlight geometries, for parameter tuning of the image processing methods.

**4.2. Detection Algorithm.** This section summarizes the image processing algorithm implemented for the light trace extraction. An example input image can be seen in Figure 6, which was captured with our test hardware (Section 3.3). The light traces, arising out of the headlights, can be clearly seen. There are some universal features of the light traces on the images, which can be utilized during the detection process. First, regardless of the vehicle and the headlight itself, it is typically a saturated, or nearly saturated, area on the image. In most cases, the headlight itself and the first section of the trace are saturated, and, depending on the headlight type, the intensity of the trace is decreasing towards its endpoint. Second, if the sensor is aligned horizontally and the camera holder is placed



FIGURE 5: A collection of the images taken from the databases.

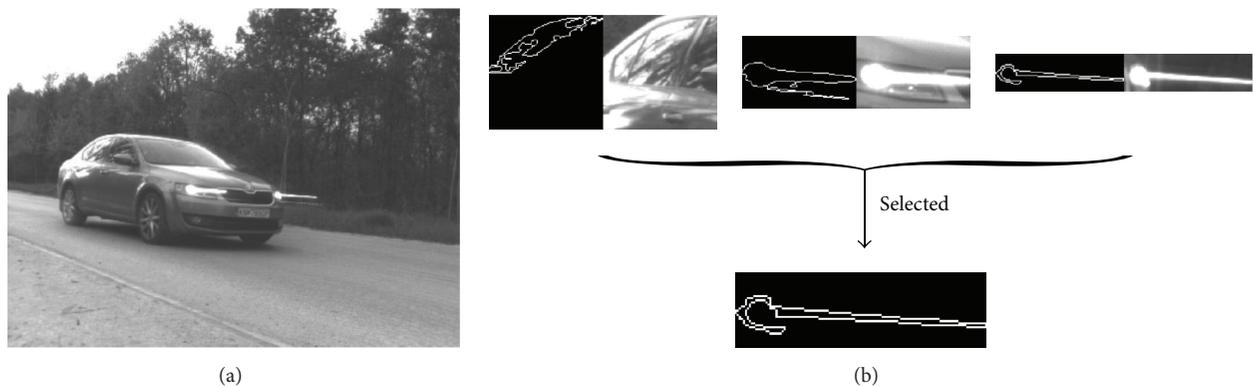


FIGURE 6: (a) Example for an input image. (b) Candidate objects and the selected light trace.

at  $\sim 0.6$  m from the ground, where the headlights are expected to be approximately, the traces will appear as horizontal edges on the images.

As a first step, we apply histogram transformation to highlight the bright regions on the image and to suppress other parts of the scene, so that less processing will take place in the irrelevant regions of the image in the later steps. This is followed by an anisotropic edge enhancement, to highlight horizontal edges. Then, thresholding is the next step. As described previously, the regions in question are typically nearly saturated; therefore, after edge enhancement, a universal high binarization threshold can be used. This results

in a binary image, from which we can extract and label blob (binary large object) boundaries. Then, we filter out blobs based on boundary length. Blobs with boundary length above and under a threshold level are discarded, and the remaining will be considered candidate objects. These minimum and maximum thresholds have been defined based on the learning image set and tested to ensure maximum reliability. On each image, we will get a number of candidate objects. If the input image was the one on the left side of Figure 6, we would get the objects as candidates indicated in the right side of Figure 6. The remaining blobs are again filtered, according to the ratio of horizontal to vertical size of their bounding box.

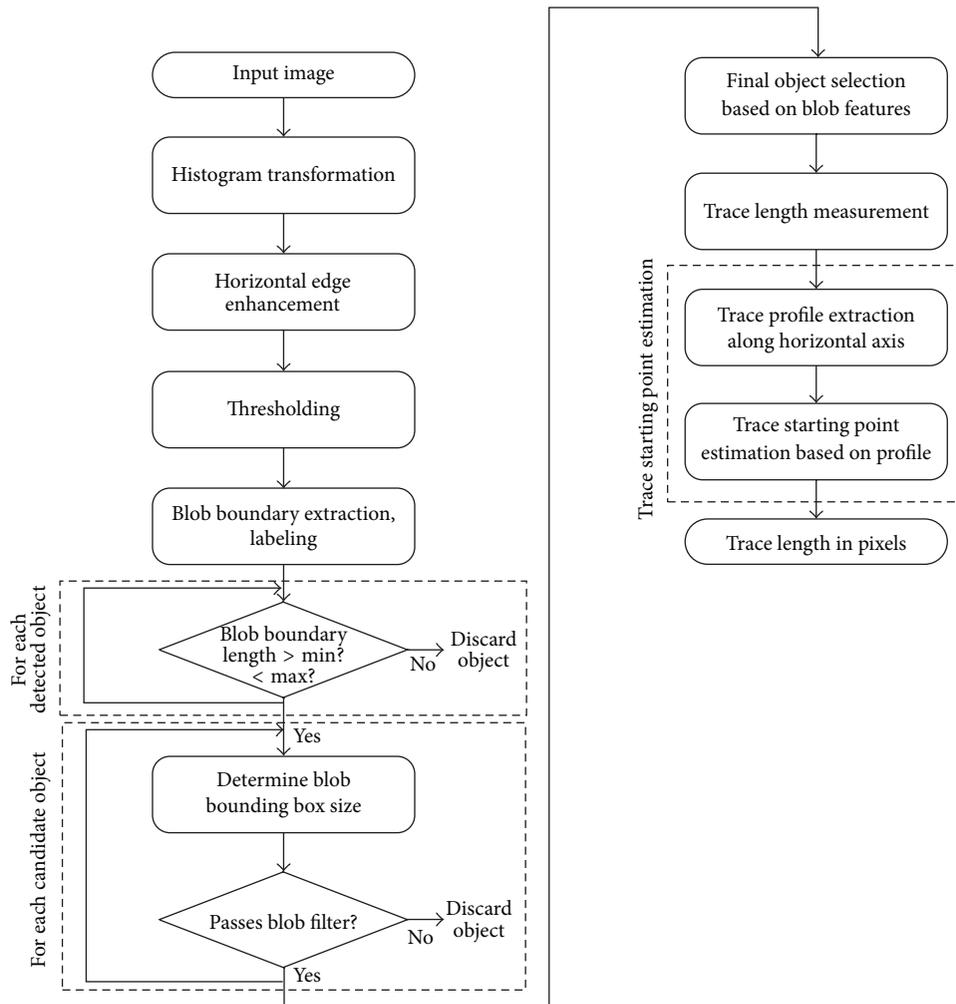


FIGURE 7: Flowchart of the algorithm.

Selection of the final object from the candidates is based on morphological features. As you can see in Figure 6, reflection on the car body can modulate the shape of the light trace which is geometrically closer to the camera, making the measurement problematic, so we always prefer the farther trace in the selection process. The output of the algorithm is the full horizontal size of the selected blob, including the saturated area of the headlight. The above described algorithm is capable of detecting the light traces at a 91.46% ratio (based on the previously mentioned evaluation image set), if the input images are captured in the previously described way. The flowchart of the algorithm is shown in Figure 7.

## 5. Trace Length Measurement and Correction

After the light trace has been detected on the input image, we have to measure its length precisely, in order to get a precise speed estimate for the movement. The output of the trace detection is the horizontal size of the selected blob (denoted with  $x$  in Figure 8). To measure the interframe movement of the headlight, we need to identify both endpoints of the

traces. Identification of the starting point of the trace is difficult, because there is a saturated area around it, as you can see in Figures 6 and 8. In this section, we summarize the methods which we developed for the trace length correction.

*5.1. Acquiring Ground Truth with a Stereo Image Pair.* As described in Section 4.2, the proposed image processing method calculates speed estimates based on some properties of saturated or nearly saturated regions of the image. As there is information loss in those areas due to the saturation, the localization of the starting point for the trace length measurement needs to be done in a different way. Consider a second auxiliary camera synchronized to the primary sensor, which applies the same exposure settings, but with a dark neutral density filter, which cuts out 90% of the incoming light. As a result, only the brightest points of the scene will appear on this second correction image (Figure 8). Our test platform is capable of synchronizing multiple cameras, where the sensor control signals are driven by an FPGA [11, 12]. With stereo correspondence methods, we can pinpoint the position of the light source, based on the compensation image.

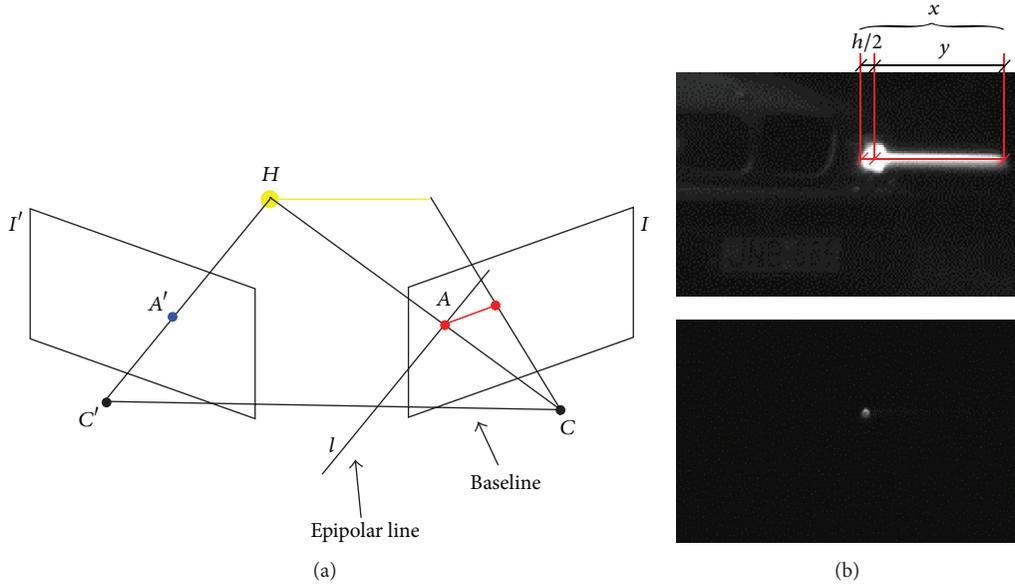


FIGURE 8: (a) Epipolar geometry in the measurement setup. (b) Original image with the notations used in trace starting point estimation and the compensation image as a result of the applied optical filter.

Let  $A$  and  $A'$  be the projections of the starting point of the trace ( $T$ ), as shown in Figure 8. The intrinsic projective geometry between the two views is defined as follows:

$$A'^T F A = 0, \quad (12)$$

where  $F$  is the fundamental matrix [14], which maps points in  $I$  to lines in  $I'$  in pixel coordinates as  $FA' = l$ , where  $l$  is the epipolar line. Consider the detected trace starting point to be a point-like object on the secondary image and the fundamental matrix of the stereo rig to be known from extrinsic and intrinsic calibration. In this case, the intersection of the epipolar line and the major axis of the detected trace (if described as a blob) defines the starting point of the trace on  $I$ . After that, the length of the trace can be measured. Later on, we consider this as the ground truth. As in most cases the saturated region on the compensation image is a point-like object with good approximation, and uncertainty caused by the size of the detected blob on the secondary image is negligible.

To verify the obtained results, we applied an Inertial Measurement Unit (IMU) on a vehicle to log the speed of the car in a real situation. Our solution offers 1.3% error compared to the IMU measurement, which encourages us to use this stereo method for acquiring the ground truth. The description of the proof-of-concept measurement and the related figures can be found in [15].

**5.2. Statistical Trace Starting Point Localization Based on a Single Camera.** When using a single camera for capturing the images, the best possible option could be a statistical based estimation of the starting point of the traces for the trace

length measurement. According to Figure 8, we estimate the length of the traces in the following way:

$$y = x - \frac{h}{2}, \quad (13)$$

where the  $x$  is the length of the detected blob,  $y$  is the trace itself, and  $h$  is the horizontal size of the headlight, respectively. This is based on an assumption that the starting point of the trace is in the middle of the headlight. For this calculation, we developed an algorithm to separate the beam originating from the headlight and the saturated region of the headlight itself, based on the vertical profile of the detected blobs along the horizontal axis. According to our stereo database, the mean value of the difference between the calculated headlight center and the light trace starting point obtained through the stereo correspondence method is 3.2 pixels, and the deviation is 1.6 pixels. Using the described method for trace starting point estimation, we ran the trace detection algorithm and evaluated the results compared to the ground truth. Figure 9 shows the error of the detection method. The whole detection and measurement algorithm using only a single double-exposed frame of a single camera resulted in 4.1% overall accuracy. As a result, it could be used, for example, as a sensing node of a smart city sensor network for traffic surveillance and monitoring, but not for precise speed measurement.

As calculated trace length is proportional to movement speed and inversely proportional to the distance between the camera and the vehicle, less distance between the vehicle and the camera and higher movement speed mean higher accuracy. Notice that, in Figure 9, different error values can be observed for samples with similar trace length. This effect comes from the difference between the estimated and the real headlight center in the case of different headlight geometries.

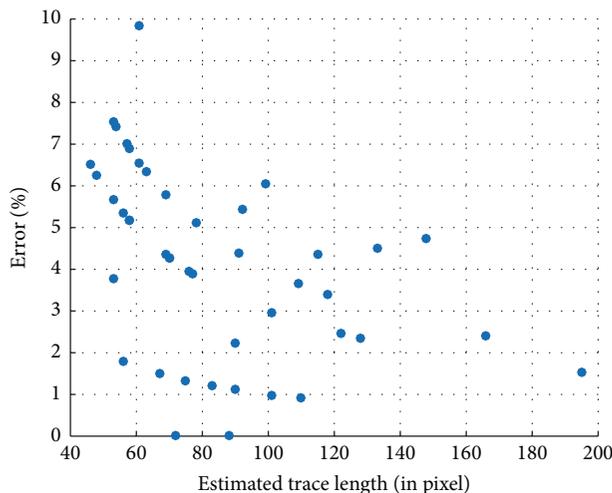


FIGURE 9: Error of the estimated trace length compared to the ground truth obtained through stereo correspondence, with respect to estimated trace length. Notice the decreasing tendency with the estimated length. As the trace length becomes longer, the uncertainty caused by the light source localization becomes less significant compared to the total trace length. There is another interesting feature: different error values can be observed for samples with similar trace length. This is caused by the differences in headlight geometries of different vehicles, because the position of the light source inside the headlight varies with different car types.

**5.3. Accuracy Improvement Possibilities Based on a Novel Sensor Design Concept.** In this subsection, a slightly improved exposure-control scheme is proposed to improve the accuracy and reliability of the measurement method. With a novel pixel architecture and the modification of the shutter cycle, inserting one additional short close state after the primary exposure [open, close, semiopen, and close], one can achieve an image, where the light trace is separated from the saturated areas of the headlight, which greatly simplifies the measurement of its length and makes it much more accurate. This method would require a dual-pixel sensor architecture with a truly controllable shutter as well as a modified in-pixel charge storage approach. Hence, the aim of future research is to develop a custom VLSI design, capable of this separation on a hardware level.

## 6. Conclusion

To summarize the results, a novel vision based speed estimation method was developed, capable of measuring speed of specified objects based on a single double-exposed image of a single imager sensor. The measurement results are encouraging, because the published intraframe speed measurement solution [3] reached 5% accuracy in average in outdoor environment. The method presented in that paper is based on assumptions which requires high quality, high frame-rate, hence expensive cameras. Our solution offers similar accuracy with a low-end sensor and much better accuracy with a stereo pair, which can match the requirements of a speed cam sensor in good lighting conditions.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The support of the KAP-1.5-14/006 Grant and the advices of Laszlo Orzo are greatly acknowledged.

## References

- [1] D. Sawicki, *Traffic Radar Handbook: A Comprehensive Guide to Speed Measuring Systems*, Authorhouse, Bloomington, Ind, USA, 2002.
- [2] F. W. Cathey and D. J. Dailey, "A Novel Technique to Dynamically Measure Vehicle Speed Using Uncalibrated Roadway Cameras," 2015, <http://www.its.washington.edu/pubs/AutoCamCal-IV05-2.pdf>.
- [3] H.-Y. Lin, K.-J. Li, and C.-H. Chang, "Vehicle speed detection from a single motion blurred image," *Image and Vision Computing*, vol. 26, no. 10, pp. 1327–1337, 2008.
- [4] H.-Y. Lin and C.-H. Chang, "Automatic speed measurements of spherical objects using an off-the-shelf digital camera," in *Proceedings of the IEEE International Conference on Mechatronics (ICM '05)*, pp. 66–71, Taipei, Taiwan, July 2005.
- [5] M. Celestino and O. Horikawa, "Velocity measurement based on image blur," *ABCM Symposium Series in Mechatronics*, vol. 3, pp. 633–642, 2008.
- [6] O. Yadid-Pecht and R. Etienne-Cummings, *CMOS Imagers From Phototransduction to Image Processing*, Kluwer Academic, 2004.
- [7] P. B. Catrysse and B. A. Wandell, "Optical efficiency of image sensor pixels," *Journal of the Optical Society of America A*, vol. 19, no. 8, pp. 1610–1620, 2002.
- [8] S. Velichko, G. Agranov, J. Hyneczek et al., "Low noise high efficiency 3.75  $\mu\text{m}$  and 2.8  $\mu\text{m}$  global shutter CMOS pixel arrays," in *Proceedings of the International Image Sensor Workshop (IISW '13)*, Snowbird, Utah, USA, June 2013.
- [9] S. Lauthermann, A. Lee, J. Stevens, and A. Joshi, "Comparison of global shutter pixels for CMOS image sensors," in *Proceedings of the International Image Sensor Workshop (IISW '07)*, Ogunquit, Me, USA, June 2007.
- [10] J. Solhusvik, S. Velichko, T. Willassen et al., "A 1.2MP 1/3" global shutter CMOS image sensor with pixel-wise automatic gain selection," in *Proceedings of the International Image Sensor Workshop (IISW '11)*, Hokkaido, Japan, June 2011.
- [11] Á. Zarándy, M. Nemeth, Z. Nagy, A. Kiss, L. Santha, and T. Zsedrovits, "A real-time multi-camera vision system for UAV collision warning and navigation," *Journal of Real-Time Image Processing*, 2014.
- [12] A. Zarandy, Z. Nagy, B. Vanek, T. Zsedrovits, A. Kiss, and M. Nemeth, "A five-camera vision system for UAV visual attitude calculation and collision warning," in *Computer Vision Systems: 9th International Conference, ICVS 2013, St. Petersburg, Russia, July 16–18, 2013. Proceedings*, vol. 7963 of *Lecture Notes in Computer Science*, pp. 11–20, Springer, Berlin, Germany, 2013.
- [13] US Department of Transportation National Highway Traffic Safety Administration, "Speed-Measuring Device Performance Specifications," 2015, <http://www.theiacp.org/portals/0/pdfs/IACPLidarModule.pdf>.

- [14] J. Mallon and P. F. Whelan, "Projective rectification from the fundamental matrix," *Image and Vision Computing*, vol. 23, no. 7, pp. 643–650, 2005.
- [15] M. Nemeth and A. Zarandy, "New sensor concept for intra-frame scene and speed capturing," in *Proceedings of the European Conference on Circuit Theory and Design (ECCTD '15)*, pp. 1–4, IEEE, Trondheim, Norway, August 2015.

## Research Article

# Night-Time Vehicle Sensing in Far Infrared Image with Deep Learning

Hai Wang,<sup>1</sup> Yingfeng Cai,<sup>2</sup> Xiaobo Chen,<sup>2</sup> and Long Chen<sup>2</sup>

<sup>1</sup>*School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China*

<sup>2</sup>*Automotive Engineering Research Institute, Jiangsu University, Zhenjiang 212013, China*

Correspondence should be addressed to Yingfeng Cai; [caicaixiao0304@126.com](mailto:caicaixiao0304@126.com)

Received 27 May 2015; Accepted 3 August 2015

Academic Editor: Fadi Dornaika

Copyright © 2016 Hai 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.

The use of night vision systems in vehicles is becoming increasingly common. Several approaches using infrared sensors have been proposed in the literature to detect vehicles in far infrared (FIR) images. However, these systems still have low vehicle detection rates and performance could be improved. This paper presents a novel method to detect vehicles using a far infrared automotive sensor. Firstly, vehicle candidates are generated using a constant threshold from the infrared frame. Contours are then generated by using a local adaptive threshold based on maximum distance, which decreases the number of processing regions for classification and reduces the false positive rate. Finally, vehicle candidates are verified using a deep belief network (DBN) based classifier. The detection rate is 93.9% which is achieved on a database of 5000 images and video streams. This result is approximately a 2.5% improvement on previously reported methods and the false detection rate is also the lowest among them.

## 1. Introduction

On average, at least one person globally dies in a vehicle crash every minute. Auto accidents also injure at least ten million people each year, with two or three million of these people seriously injured [1]. To address this problem, Advanced Driver Assistant Systems (ADAS) are used more and more often to provide assistance and supplementary information for drivers. Current ADAS developments include many functions such as lane departure warning, forward collision warning, parking assistance systems, and night vision enhancement [2].

The vision sensor is one of the most popular sensors in ADAS and many algorithms are designed to use it including vehicle detection, lane detection, pedestrian detection, and traffic sign recognition [3–6]. Among these methods, vehicle detection is a popular research area. Monocular vision is often used in this task. The Sivaraman model considers vehicle detection as a two-class classification problem and trains an Adaboost classifier with a Haar feature [7]. Stereo vision is another common method. Hermes et al. use stereo vision and propose a vehicle detection method based on

density map clustering [8]. Additionally, motion based algorithms have also been used for overtaking vehicle detection [9].

The current reality is that a large number of road accidents occur during times of low visibility such as at night time. Statistical data also demonstrates that more than half of accidents causing fatalities occur at night. However, most existing vehicle detection systems and algorithms are more focused on daylight vehicle detection with visible spectrum cameras. Although some researchers have put effort into night-time vehicle detection with vehicle lamp detection, the detection results are often affected by many factors such as low illumination, light reflection on rainy days, and the camera exposure time [10]. To compensate for these limitations of visible spectrum cameras, far infrared (FIR) sensors are becoming more and more frequently employed for night-time vehicle detection tasks. They do not require any illumination source and rely purely on heat signatures from the environment to produce a gray scale image. Therefore, FIR sensors can detect the infrared heat signatures generated by vehicle parts such as the engine, wheels, and exhaust pipe. A group of typical images captured at night



FIGURE 1: Night views using visible spectra (a) and FIR spectrum cameras (b).

by a visible spectrum camera and FIR camera is shown in Figure 1.

Compared with the visible spectra camera, the image captured by the FIR spectrum camera lacks color and detailed information. Up until now, not much work has been done specifically on vehicle detection in FIR images. In 2000, Andreone (University of Parma) designed an FIR camera based vehicle detection prototype car and detected vehicles by mainly relying on size, shape, and the content distribution of the image high intensity area [11]. In 2010, Besbes et al. introduced a machine learning framework to this task and designed a vehicle detection method with SURF-based features and an SVM classifier [12]. There are also many researchers that focus on the similar problem of detecting pedestrians in an FIR image. Most of the proposed methods use a two-class classification framework and use different features such as HOG and different classifiers such as SVM and Adaboost [13, 14].

A review of the approaches proposed in the literature finds that vehicle detection in an FIR image is most commonly performed using a two-class classification framework. However, a newly proposed approach in pattern recognition named deep learning has also been used in rare studies. Classifiers such as SVM and Adaboost are all shallow learning models because they can be modeled as a structure with one input layer, one hidden layer, and one output layer. Deep learning refers to a class of machine learning techniques which exploit hierarchical architectures for representation learning and pattern classification. In contrast to shallow models, deep learning has the ability to learn multiple levels of representation and abstraction that enables better understanding of the image data. Deep belief networks (DBN) model is a typical deep learning structure which is first proposed by Hinton and has demonstrated its success in simple image classification tasks of MNIST [15].

In this work, a deep learning based vehicle detection algorithm for FIR images is proposed. In Section 2, the vehicle candidate generation method in FIR image will be described. Candidate shape segmentation will then be performed with a contour generation method in Section 3. In Section 4, vehicle verification will be implemented using shape feature vectors and a deep belief network. The experiments and conclusion will be presented in Sections 5 and 6 separately.

## 2. Vehicle Candidate Generation

For vehicle detection in an image, two steps are usually performed which are vehicle candidate generation (VCG) and vehicle candidate verification (VCV) [16]. In VCG, all image areas which have any probability of being vehicles will be selected. In this step, prior knowledge of vehicles may often be used, such as horizontal/vertical edges, symmetry, color, shadow, and texture. In VCV, the image areas selected in VCG will be further verified to eliminate those which are not vehicles. In this step, a two-class classification framework is often used and a classifier that can distinguish vehicles from nonvehicles will be trained from a set of training images. In our work, we will also follow this two-step framework.

The FIR images reflect information on the temperature of objects. Therefore, FIR image cannot show detailed information available in the visible domain such as texture, color, and shadows. Obviously, the existing vehicle detection algorithms used with visible spectrum cameras are not suitable for far infrared cameras because of the inherent differences between images generated by cameras in the IR and visible spectra. Due to this, vehicle candidate generation mainly focuses on hot spots in the image.

In the VCG step of our method, a low threshold is firstly applied to the pixel values. The pixels with values that are lower than the threshold are considered to be low temperature areas and are removed, while hot spots in the images are preserved. Since the relationship between the temperature of the object and the brightness in FIR image is constant for a specific FIR camera, the threshold can be chosen manually. In our application, the threshold is set at 150 which corresponds to a temperature of 30°C. Figure 2 shows the original FIR image and the processed image separately.

In the processed image, there obviously exist many hot spots that are not belonging to vehicles such as the hot road surface, hot road lamps, or other sources. To eliminate these sources of interference, a connected region searching algorithm is applied on the processed image and all regions that do not satisfy the rules below are considered to be nonvehicle hot spots:

- (1) Rule 1: connected regions with a length/width ratio below 0.3 or above 1.5 are considered to be nonvehicle hot spots.

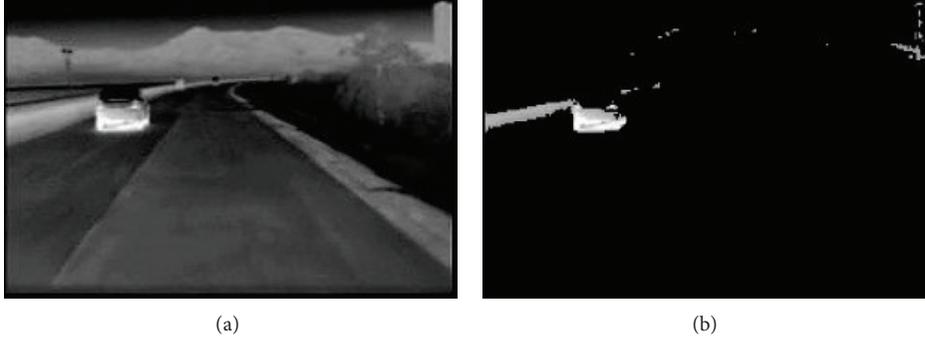


FIGURE 2: Hot area detection in an FIR image (processed image on right).

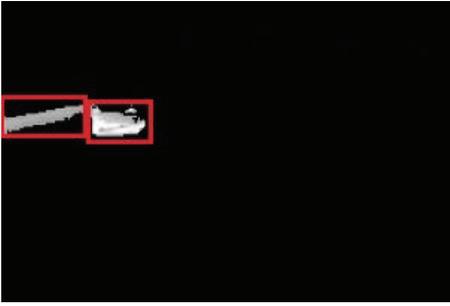


FIGURE 3: Vehicle candidate generation.

- (2) Rule 2: connected regions with less than 120 pixels are considered to be nonvehicle hot spots.

This further processing eliminates many nonvehicle hot spots. The remaining hot spots will all be considered to be vehicle candidates. As seen in Figure 3, two vehicle candidates are identified in this particular FIR image.

### 3. Vehicle Candidate Contour Segmentation

In a traditional visible spectrum image-based vehicle detection framework, a two-class classifier will be trained and used to classify all vehicle candidates that are identified. However, traditional image feature descriptors such as Haar and HOG are more suitable for local pixel information representation and in an FIR image; the detailed information is relatively small. Therefore, features such as the vehicle candidate shape or contours will be used, which are based on global information rather than pixel values. In this section, contour segmentation is firstly required to segment the full vehicle candidate objects.

One way to find continuous contours in a gray scale image is using edge detection to find continuous contours in the edge image. However, since edges are generally discrete and independent, the contours that are generated may appear broken and branching. Another method transforms the scale image to a binary image first and then uses the chain code method to generate contours in the binary image. For this method, it is critical that binary image is of good quality.

Traditional binary segmentation methods are usually based on a global threshold obtained from a pixel histogram. This type of method is suitable when the inner foreground image has small variations, such as pedestrians in an FIR image. However, for our application, the lower part of the vehicle such as the wheels is usually brighter than the upper part of the vehicle such as the windows. Therefore, a global threshold based method could easily eliminate the upper part of vehicles in an FIR image.

Based on the analysis above, a maximum distance based local adaptive threshold determination method is proposed to generate the binary segmentation. This method firstly sets a global threshold based on a histogram to get a binary image. Then, a maximum distance based local threshold will be further decided for each subregion around the edge of the binary image that has been generated. The main steps of the local threshold determination method are given below.

(1) Set the processing area of each vehicle candidate. Specifically, the original vehicle candidate areas are expanded twice vertically and 1.5 times horizontally.

(2) Apply a median filter on the vehicle candidate area to eliminate obvious noise.

(3) Set  $I(x, y)$  as the vehicle candidate area image. Set a global threshold using the OTSU method. Fill in the blank area of the binary image and get the edge image of  $I(x, y)$  which is set as  $E(x, y)$ .

(4) Choose the edge points with a five-point gap in  $E(x, y)$ . Then, set the selected edge points as the center and find the selected edge point  $5 \times 5$  neighborhood region to obtain a new local threshold for this small region, which is decided using the maximum distance based method. This method is based on the concept that a threshold should divide a gray scale histogram into two parts. The best threshold produces the largest gap between the mean value of the two parts and the mean value of the whole image. The distance measurement function is

$$R(T) = \frac{\left[ \sum_{i=1}^T iP_i - \sum_{i=1}^M iP_i P_i(T) \right]^2}{P_i(T) [1 - P_i(T)]}, \quad (1)$$

where  $M$  is the total number of gray scale levels,  $P_i$  is the proportion of gray scale level  $i$ , and  $T$  is the threshold,  $P_i(T) = \sum_{i=1}^T P_i$ .

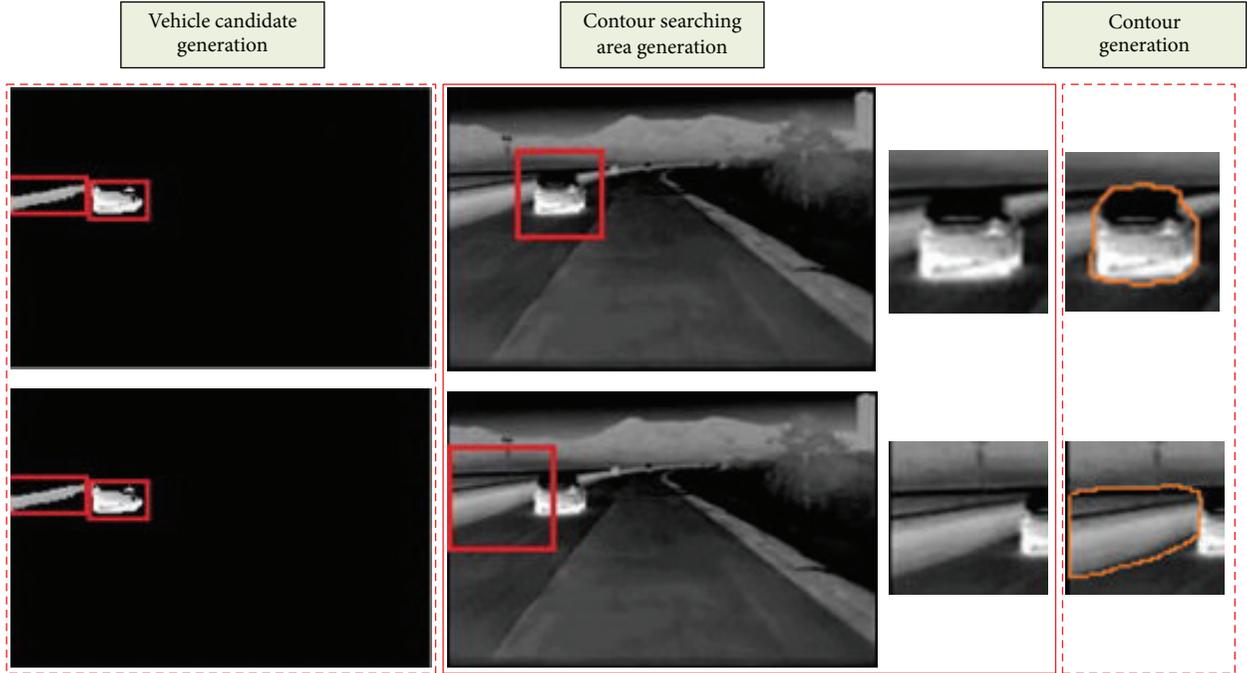


FIGURE 4: Vehicle candidate contour generation.

Then, transform the small region to a binary image with this new threshold and set the new binary image to be  $bi(x, y)$ .

(5) Repeat step (4) until the complete areas around each of the selected edge points are processed and a full contour of the vehicle candidates is obtained.

The processed result of the contour generation method is shown in Figure 4.

#### 4. Vehicle Candidate Verification

In this section, a deep belief network (DBN) based vehicle candidate verification algorithm will be proposed.

The machine learning based method is very popular and effective for vehicle candidate verification tasks in the visible spectrum. Within the many existing machine learning methods, SVM (support vector machines) and Adaboost are the two most common classifiers [17–20]. However, most classifiers including these two classifiers are based on a shallow learning model and can be modeled as a structure consisting of one input layer, one output layer, and a single hidden layer. Recently, a new machine learning structure called deep learning has been proposed, which has a hierarchical architecture that can be exploited for representation learning and pattern classification. The deep model is superior to the existing shallow models as it can learn multiple levels of representation and abstraction of image data.

There are many types of deep architectures such as deep belief networks (DBN) and deep convolution neural networks (DCNN) which use DBN as a typical deep learning structure, as first proposed by Hinton et al. [21] and used in many tasks such as MNIST classification, 3D object recognition, and voice recognition. In our work, DBN is applied

and a classifier is trained for vehicle candidate verification tasks.

*4.1. Deep Belief Network (DBN) for Vehicle Candidate Verification.* In this subsection, the overall architecture of the DBN classifier for vehicle candidate verification will be firstly introduced.

Let  $X$  be the set of training samples which contain vehicle contour images and nonvehicle contour images which are generated manually by our group.  $X$  consists of  $K$  samples which are shown below:

$$X = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_k, \dots, \mathbf{X}_K]. \quad (2)$$

In  $X$ ,  $\mathbf{X}_k$  is a training sample and all samples are resized to  $I \times J$ .  $Y$  represents the labels corresponding to  $X$ , which can be written as

$$Y = [y_1, y_2, \dots, y_k, \dots, y_K]. \quad (3)$$

In  $Y$ ,  $y_k$  is the label vector of  $\mathbf{X}_k$ . If  $\mathbf{X}_k$  belongs to a vehicle,  $y_k = (1, 0)$ . Otherwise,  $y_k = (0, 1)$ .

The purpose of the vehicle candidate verification task is to learn the mapping function from the training data  $X$  to the label data  $Y$  based on a given training set. With this trained mapping function, unknown contour images can be classified as either vehicle or nonvehicle.

In this task, a DBN architecture is applied to address this problem. Figure 5 shows the overall architecture of the DBN. It is a fully interconnected deep belief network including one visible input layer  $V^1$ ,  $N$  hidden layers  $H^1, \dots, H^N$ , and one visible label layer  $La$  at the top. The visible input layer  $V^1$  maintains a  $I \times J$  neural number which is equal to

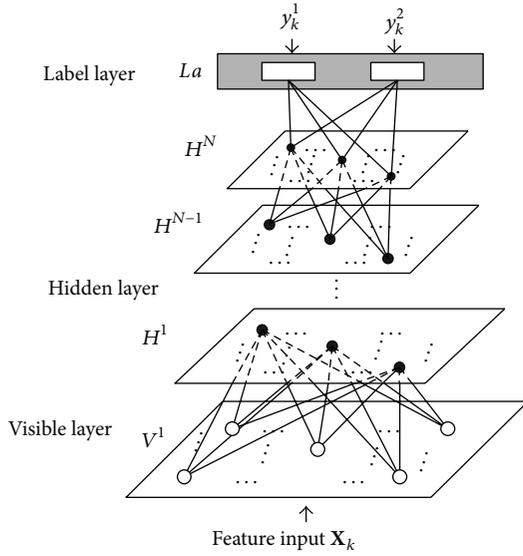


FIGURE 5: Proposed DBN for vehicle candidate verification.

the dimensions of the training feature, that is, the pixel values of the training samples. On the top, the La layer just has two states which can be either (1, 0) or (0, 1).

The learning process of the DBN has two main steps. In the first step, the parameters of the two adjacent layers will be refined with the greedy-wise reconstruction method. This step will be repeated until the parameters of all the hidden layers are fixed. This first step is also called the pretraining process. In the second step, the whole pretrained DBN will be fine-tuned with the La layer information based on back propagation. This second step can be considered to be the supervised training step.

**4.2. Pretraining Method.** In this subsection, the training method of the whole DBN for vehicle candidate verification will be presented.

Assume that the size of the upper layer is  $P \times Q$  whose size is decided in bilinear projection as mentioned in Zhong et al.'s work [22]. In this step, the parameters of the two adjacent layers will be refined using the greedy-wise reconstruction method proposed by Hinton et al. [21]. The visible input layer  $V^1$  and the first hidden layer  $H^1$  are taken here as an example; other adjacent layers use the same pretraining method.

The visible input layer  $V^1$  and the first hidden layer  $H^1$  contract a Restrict Boltzmann Machine (RBM).  $I \times J$  is the neural number in  $V^1$  and  $P \times Q$  is that of  $H^1$ . The energy of state  $(\mathbf{v}^1, \mathbf{h}^1)$  in this RBM is

$$\begin{aligned}
 E(\mathbf{v}^1, \mathbf{h}^1, \theta^1) &= -(\mathbf{v}^1 \mathbf{A} \mathbf{h}^1 + \mathbf{b}^1 \mathbf{v}^1 + \mathbf{c}^1 \mathbf{h}^1) \\
 &= - \sum_{i=1, j=1}^{i \leq I, j \leq J} \sum_{p=1, q=1}^{p \leq P, q \leq Q} v_{ij}^1 A_{ij,pq}^1 h_{pq}^1 \\
 &\quad - \sum_{i=1, j=1}^{i \leq I, j \leq J} b_{ij}^1 v_{ij}^1 - \sum_{p=1, q=1}^{p \leq P, q \leq Q} c_{pq}^1 h_{pq}^1,
 \end{aligned} \quad (4)$$

where  $\theta^1 = (\mathbf{A}^1, \mathbf{b}^1, \mathbf{c}^1)$  are the parameters between the visible input layer  $V^1$  and the first hidden layer  $H^1$ ,  $A_{ij,pq}^1$  are the symmetric weights from input neural  $(i, j)$  in  $V^1$  to the hidden neural  $(p, q)$  in  $H^1$ , and  $b_{ij}^1$  and  $c_{pq}^1$  are the  $(i, j)$ th and  $(p, q)$ th bias of  $V^1$  and  $H^1$ . So, the RBM has the following joint distribution:

$$P(\mathbf{v}^1, \mathbf{h}^1; \theta^1) = \frac{1}{Z} e^{-E(\mathbf{v}^1, \mathbf{h}^1; \theta^1)} = \frac{e^{-E(\mathbf{v}^1, \mathbf{h}^1; \theta^1)}}{\sum_{\mathbf{v}^1} \sum_{\mathbf{h}^1} e^{-E(\mathbf{v}^1, \mathbf{h}^1; \theta^1)}}. \quad (5)$$

Here,  $Z$  is the normalization parameter and the probability that  $\mathbf{v}^1$  is assigned to  $V^1$  of this model is

$$P(\mathbf{v}^1) = \frac{1}{Z} \sum_{\mathbf{h}^1} e^{-E(\mathbf{v}^1, \mathbf{h}^1; \theta^1)} = \frac{\sum_{\mathbf{h}^1} e^{-E(\mathbf{v}^1, \mathbf{h}^1; \theta^1)}}{\sum_{\mathbf{v}^1} \sum_{\mathbf{h}^1} e^{-E(\mathbf{v}^1, \mathbf{h}^1; \theta^1)}}. \quad (6)$$

Then, the conditional distribution over the visible input state  $\mathbf{v}^1$  in layer  $V^1$  and the hidden state  $\mathbf{h}^1$  in  $H^1$  can be given by the logistic function:

$$\begin{aligned}
 p(\mathbf{h}^1 | \mathbf{v}^1) &= \prod_{p,q} p(h_{pq}^1 | \mathbf{v}^1), p(h_{pq}^1 | \mathbf{v}^1) \\
 &= \sigma \left( \sum_{i=1, j=1}^{i \leq I, j \leq J} v_{ij}^1 A_{ij,pq}^1 + c_{pq}^1 \right), \\
 p(\mathbf{v}^1 | \mathbf{h}^1) &= \prod_{i,j} p(v_{ij}^1 | \mathbf{h}^1), p(v_{ij}^1 | \mathbf{h}^1) \\
 &= \sigma \left( \sum_{p=1, q=1}^{p \leq P, q \leq Q} h_{pq}^1 A_{ij,pq}^1 + b_{ij}^1 \right),
 \end{aligned} \quad (7)$$

where  $\sigma(x) = 1/(1 + \exp(-x))$ .

Finally, the weights and biases can be updated step by step using random Gaussian distribution values  $A_{ij,pq}^1(0)$ ,  $b_{ij}^1(0)$ , and  $c_{pq}^1(0)$  with the Contrastive Divergence algorithm [23]. The updating formulas are

$$\begin{aligned}
 A_{ij,pq}^1 &= \mathcal{G} A_{ij,pq}^1 \\
 &\quad + \varepsilon_A (\langle v_{ij}^1(0) h_{pq}^1(0) \rangle_{\text{data}} - \langle v_{ij}^1(t) h_{pq}^1(t) \rangle_{\text{recon}}), \\
 b_{ij}^1 &= \mathcal{G} b_{ij}^1 + \varepsilon_b (v_{ij}^1(0) - v_{ij}^1(t)), \\
 c_{pq}^1 &= \mathcal{G} c_{pq}^1 + \varepsilon_c (h_{pq}^1(0) - h_{pq}^1(t)).
 \end{aligned} \quad (8)$$

Here,  $\langle \cdot \rangle_{\text{data}}$  represents the expectation with respect to the data distribution and  $\langle \cdot \rangle_{\text{recon}}$  represents the reconstruction distribution after one step. The step size  $t$  is set to 1.

As mentioned previously, the whole pretraining process will be performed on lower layer groups ( $V^1, H^1$ ) to upper layer groups ( $H^{n-1}, H^n$ ) one at a time.

**4.3. Global Fine-Tuning.** In the above unsupervised pretraining process, the greedy layer-wise algorithm is used to



FIGURE 6: SAT NV628 FIR camera.

learn the DBN parameters. In this subsection, a traditional back propagation algorithm will be used to fine-tune the parameters  $\theta = [\mathbf{A}, \mathbf{b}, \mathbf{c}]$  using the information of the label layer  $L_a$ .

Since the pretraining process has already identified strong initial parameters, the back propagation step is just used to finely adjust the parameters so that local optimum parameters  $\theta^* = [\mathbf{A}^*, \mathbf{b}^*, \mathbf{c}^*]$  can be found. At this stage, the learning objective is to minimize the classification error  $[-\sum_t y_t \log \hat{y}_t]$ , where  $y_t$  and  $\hat{y}_t$  are the real label and the output label of data  $\mathbf{X}_t$  in layer  $N$ .

## 5. Experiments and Analysis

*5.1. DBN Based Vehicle Verification Effect.* The proposed DBN based vehicle verification method is trained on our image dataset captured by a SAT NV628 FIR camera as shown in Figure 6. The total number of samples for training and testing are 2700 and 500, respectively.

By using the proposed method, four different architectures of 2D-DBN are applied. They all contain one visible layer and one label layer, but with one, two, three, and four hidden layers, respectively. In training, the critical parameters of the proposed 2D-DBN in experiments are set to  $\alpha = 0.4$  and  $\vartheta = 0.75$  and the image samples for training are all resized to  $32 \times 32$  pixels.

The detection results of the four 2D-DBN architectures and two common shallow models (SVM and Adaboost) are shown in Table 1. It is observed that the 2D-DBN with three hidden layers maintains the highest detection rate in the test set. It is also seen that deep architecture performs much better than the existing shallow models.

*5.2. System Overall Effect.* All the methods described below are tested using the same image dataset containing 5000 images captured by our group. In the dataset, there are 6382 vehicles and around 37% in near range (less than 25 m), 61% in medium range (25 m to 75 m), and 12% in far range (more than 75 m). Some of the vehicle candidate generation effects are shown in left column of Figure 7. Based on the

TABLE 1: Vehicle detection rates with different DBN structures and two shallow models.

Classifier types	Correct detection	Correct rate
2D-DBN (one hidden layer)	455/500	91.0%
2D-DBN (two hidden layers)	469/500	93.8%
2D-DBN (three hidden layers)	474/500	94.8%
2D-DBN (four hidden layers)	468/500	93.60%
HOG + SVM	438/500	87.6%
Haar + Adaboost	442/500	88.4%

TABLE 2: Overall vehicle detection effects comparison.

Authors	TP rate	FP rate	Notes
Andreone et al. [11]	85.8%	8.70%	30 Hz frame rate
Besbes et al. [12]	91.4%	3.48%	24 Hz frame rate
<b>Our framework</b>	<b>93.9%</b>	<b>1.13%</b>	21 Hz frame rate

vehicle candidate generation results, the DBN based vehicle candidate verification method is further applied. Some of the vehicle candidate verification results are shown in Figure 7. The left column shows identified vehicle candidates marked in red and the right column shows the verified vehicles marked with a blue rectangle.

The overall vehicle detection effects are shown in Table 2, as well as some state-of-the-art vehicle detection effects.

From the results shown in Table 2, it is seen that the proposed vehicle detection framework exhibits the lowest false positive (FP) rate while achieving the highest true positive (TP) rate, which is 2.5% higher than that of Bassem's method.

Figure 8 shows a group of vehicle sensing results in a continuous video. The blue rectangles represent correct vehicle detection and the red rectangles represent missed detection or false detection. From the results, it can be seen that some light spots with similar shapes to vehicles are recognized to be vehicles, such as in the fourth image. Vehicles which are not in the rear view are easily missed. Besides, the strong occlusion between different vehicles easily causes missed detection because they may be decided as one object in vehicle candidate generation step. Generally, most vehicles in the rear view are detected correctly and pedestrians and bicycles are not falsely detected.

## 6. Conclusion

In this work, a new method is proposed for night-time vehicle detection in far infrared images. Compared with existing methods, a maximum distance based local adaptive threshold determination method is proposed to generate the vehicle candidate and a deep learning framework is introduced to perform vehicle candidate verification. Overall, this two-step vehicle detection method achieves the highest vehicle detection rate compared with existed state-of-the-art methods. Additionally, the processing time is below 50 ms per frame which satisfies requirements for real-time applications.

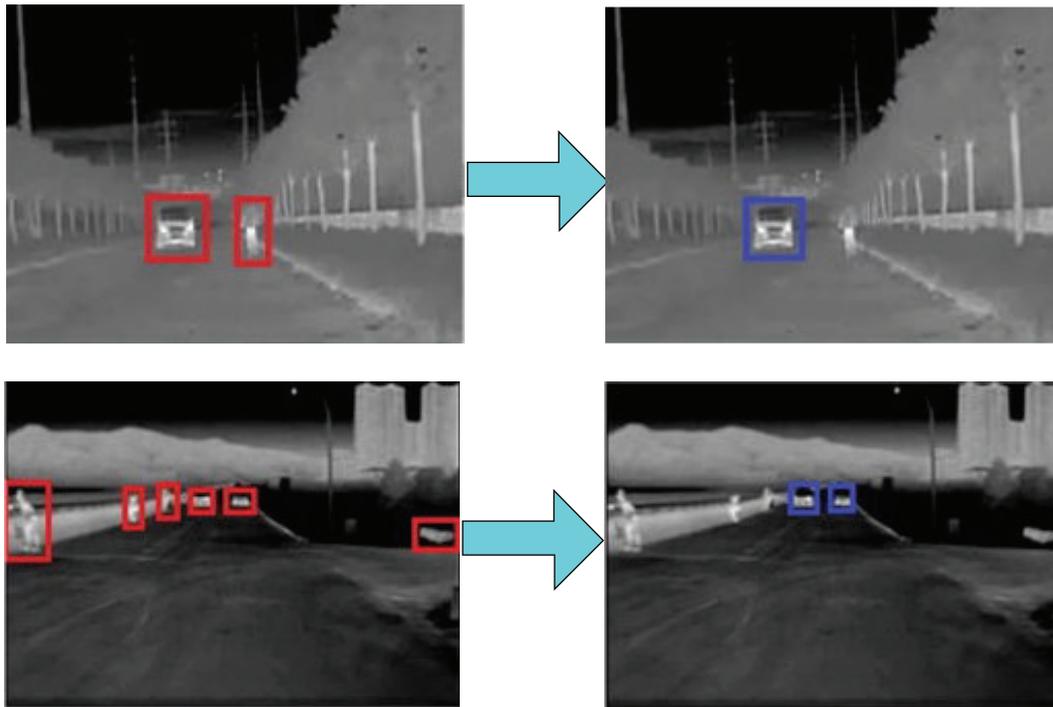


FIGURE 7: Vehicle candidate verification results.

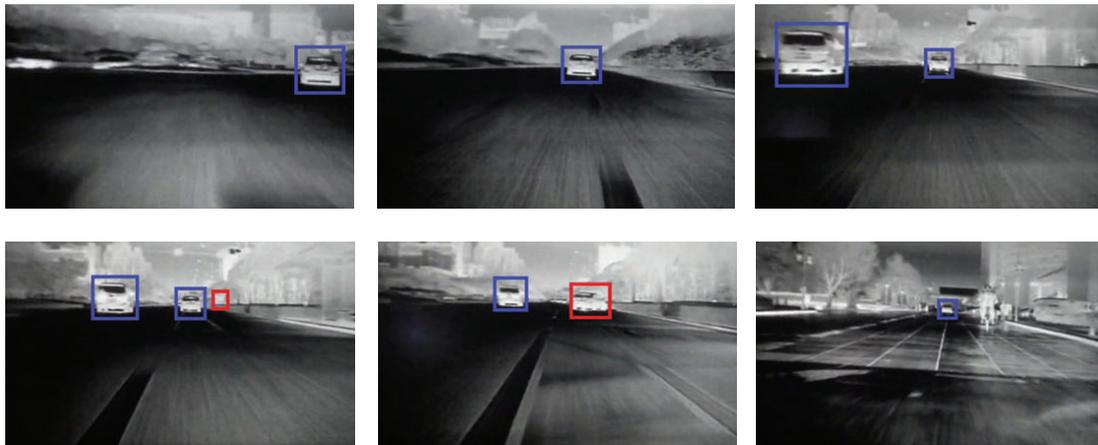


FIGURE 8: Vehicle sensing results in a continuous video.

### Conflict of Interests

The authors declare that they have no conflict of interests.

### Acknowledgments

This work has been supported by the National Natural Science Foundation of China under the Grants 61403172, 61203244, and 51305167, China Postdoctoral Science Foundation (2014M561592), China Postdoctoral Science Foundation Special Funding (2015T80511), Information Technology Research Program of Transport Ministry of China under the Grant 2013364836900, Natural Science Foundation of Jiangsu Province (BK20140555), and Jiangsu University Scientific

Research Foundation for Senior Professionals (12JDG010, 14JDG028).

### References

- [1] <http://asirt.org/Initiatives/Informing-Road-Users/Road-Safety-Facts/Road-Crash-Statistics>.
- [2] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: a survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, 2013.
- [3] H. Wang, C. Yuan, and Y. Cai, "Smart road vehicle sensing system based on monocular vision," *Optik*, vol. 126, no. 4, pp. 386–390, 2015.

- [4] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine Vision and Applications*, vol. 25, no. 3, pp. 727–745, 2014.
- [5] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?" in *Computer Vision—ECCV 2014 Workshops*, vol. 8926 of *Lecture Notes in Computer Science*, pp. 613–627, Springer, 2015.
- [6] F. Zaklouta and B. Stanculescu, "Real-time traffic sign recognition in three stages," *Robotics and Autonomous Systems*, vol. 62, no. 1, pp. 16–24, 2014.
- [7] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 267–276, 2010.
- [8] C. Hermes, J. Einhaus, M. Hahn, C. Wöhler, and F. Kummert, "Vehicle tracking and motion prediction in complex urban scenarios," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '10)*, pp. 26–33, IEEE, June 2010.
- [9] A. Ramirez, E. Ohn-Bar, and M. Trivedi, "Integrating motion and appearance for overtaking vehicle detection," in *Proceedings of the 25th IEEE Intelligent Vehicles Symposium*, pp. 96–101, IEEE, June 2014.
- [10] J. Arróspide, L. Salgado, M. Nieto, and F. Jaureguizar, "On-board robust vehicle detection and tracking using adaptive quality evaluation," in *Proceedings of the 15th IEEE International Conference on Image Processing (ICIP '08)*, pp. 2008–2011, IEEE, San Diego, Calif, USA, October 2008.
- [11] L. Andreone, P. Antonello, M. Bertozzi, A. Broggi, A. Fascioli, and D. Ranzato, "Vehicle detection and localization in infra-red images," in *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*, pp. 141–146, Singapore, 2002.
- [12] B. Besbes, A. Apatean, A. Rogozan, and A. Bensrhair, "Combining SURF-based local and global features for road obstacle recognition in far infrared images," in *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC '10)*, pp. 1869–1874, September 2010.
- [13] P. Hurney, P. Waldron, F. Morgan, E. Jones, and M. Glavin, "Night-time pedestrian classification with histograms of oriented gradients-local binary patterns vectors," *IET Intelligent Transport Systems*, vol. 9, no. 1, pp. 75–85, 2015.
- [14] Q. Liu, J. Zhuang, and S. Kong, "Detection of pedestrians at night time using learning-based method and head validation," in *Proceedings of the IEEE International Conference on Imaging Systems and Techniques (IST '12)*, pp. 398–402, IEEE, July 2012.
- [15] O. Pauplin and J. Jiang, "DBN-based structural learning and optimisation for automated handwritten character recognition," *Pattern Recognition Letters*, vol. 33, no. 6, pp. 685–692, 2012.
- [16] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 694–711, 2006.
- [17] S. Sivaraman and M. M. Trivedi, "Active learning for on-road vehicle detection: a comparative study," *Machine Vision and Applications*, vol. 25, no. 3, pp. 599–611, 2014.
- [18] S. S. Teoh and T. Bräunl, "Symmetry-based monocular vehicle detection system," *Machine Vision and Applications*, vol. 23, no. 5, pp. 831–842, 2012.
- [19] R. Sindoori, K. S. Ravichandran, and B. Santhi, "Adaboost technique for vehicle detection in aerial surveillance," *International Journal of Engineering & Technology*, vol. 5, no. 2, pp. 765–769, 2013.
- [20] J. Cui, F. Liu, Z. Li, and Z. Jia, "Vehicle localisation using a single camera," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '10)*, pp. 871–876, June 2010.
- [21] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [22] S. Zhong, Y. Liu, and Y. Liu, "Bilinear deep learning for image classification," in *Proceedings of the 19th ACM International Conference on Multimedia*, pp. 343–352, ACM, 2011.
- [23] F. Wood and G. E. Hinton, "Training products of experts by minimizing contrastive divergence," Tech. Rep., Brown University, Providence, RI, USA, 2012.