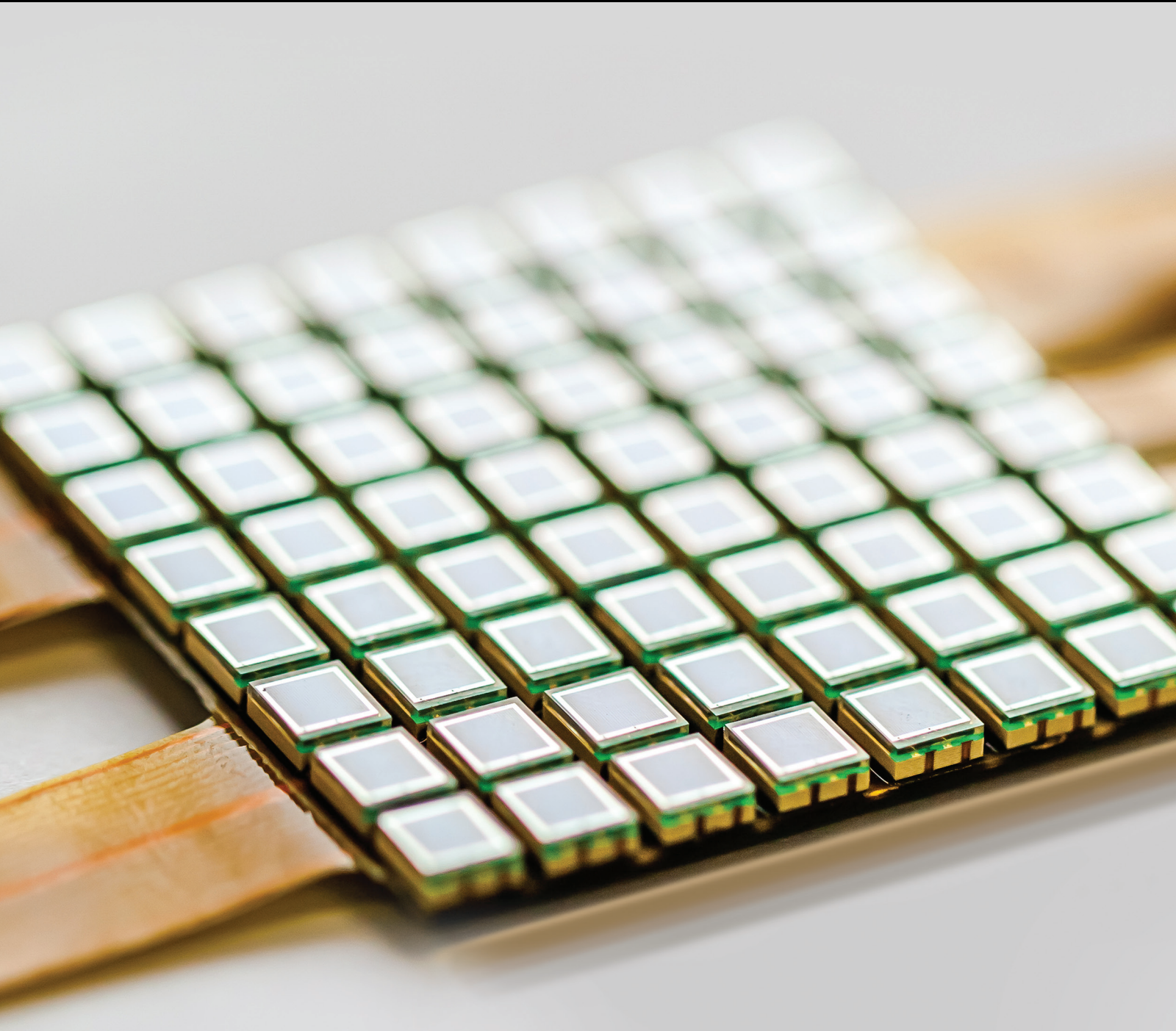


# Robotic Perception of the Sight and Touch to Interact with Environments

Guest Editors: Pablo Gil, Youcef Mezouar, Markus Vincze, and Juan A. Corrales





---

# **Robotic Perception of the Sight and Touch to Interact with Environments**



## **Robotic Perception of the Sight and Touch to Interact with Environments**

Guest Editors: Pablo Gil, Youcef Mezouar, Markus Vincze, and Juan A. Corrales



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  
Bruno Andò, Italy  
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 Dao, Australia  
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  
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  
Kourosh 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  
Jaime Lloret, Spain  
Yu-Lung Lo, Taiwan  
Oleg Lupan, Moldova  
Frederick Mailly, 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, UK  
Raymond Swartz, USA  
Hidekuni Takao, Japan  
Isao Takayanagi, Japan  
Guiyun Tian, UK  
Suna Timur, Turkey  
Hana Vaisocherova, Czech Republic  
Qihao Weng, USA  
Matthew J. Whelan, USA  
Hai Xiao, USA



# Contents

---

## **Robotic Perception of the Sight and Touch to Interact with Environments**

Pablo Gil, Youcef Mezouar, Markus Vincze, and Juan A. Corrales

Volume 2016, Article ID 1751205, 2 pages

## **A Study of Visual Descriptors for Outdoor Navigation Using Google Street View Images**

L. Fernández, L. Payá, O. Reinoso, L. M. Jiménez, and M. Ballesta

Volume 2016, Article ID 1537891, 12 pages

## **Feature Selection for Intelligent Firefighting Robot Classification of Fire, Smoke, and Thermal Reflections Using Thermal Infrared Images**

Jong-Hwan Kim, Seongsik Jo, and Brian Y. Lattimer

Volume 2016, Article ID 8410731, 13 pages

## **Monte Carlo Registration and Its Application with Autonomous Robots**

Christian Rink, Simon Kriegel, Daniel Seth, Maximilian Denninger,

Zoltan-Csaba Marton, and Tim Bodenmüller

Volume 2016, Article ID 2546819, 28 pages

## **Underwater Object Tracking Using Sonar and USBL Measurements**

Filip Mandić, Ivor Rendulić, Nikola Mišković, and Đula Nad

Volume 2016, Article ID 8070286, 10 pages

## **Robotic Visual Tracking of Relevant Cues in Underwater Environments with Poor Visibility Conditions**

Alejandro Maldonado-Ramírez and L. Abril Torres-Méndez

Volume 2016, Article ID 4265042, 16 pages

## **Vision-Based Autonomous Underwater Vehicle Navigation in Poor Visibility Conditions Using a Model-Free Robust Control**

Ricardo Pérez-Alcocer, L. Abril Torres-Méndez, Ernesto Olguín-Díaz,

and A. Alejandro Maldonado-Ramírez

Volume 2016, Article ID 8594096, 16 pages

## **Indoor Positioning System Using Depth Maps and Wireless Networks**

Jaime Duque Domingo, Carlos Cerrada, Enrique Valero, and J. A. Cerrada

Volume 2016, Article ID 2107872, 8 pages

## **Cable Crosstalk Suppression in Resistive Sensor Array with 2-Wire S-NSDE-EP Method**

JianFeng Wu and Lei Wang

Volume 2016, Article ID 8051945, 9 pages

## Editorial

# Robotic Perception of the Sight and Touch to Interact with Environments

**Pablo Gil,<sup>1</sup> Youcef Mezouar,<sup>2</sup> Markus Vincze,<sup>3</sup> and Juan A. Corrales<sup>2</sup>**

<sup>1</sup>Physics, Systems Engineering and Signal Theory Department, University of Alicante, 03690 San Vicente del Raspeig, Spain

<sup>2</sup>Université Blaise Pascal, SIGMA'Clermont, Institut Pascal, 63000 Clermont-Ferrand, France

<sup>3</sup>Automation and Control Institute, Technical University Vienna, 1040 Vienna, Austria

Correspondence should be addressed to Pablo Gil; [pablo.gil@ua.es](mailto:pablo.gil@ua.es)

Received 14 November 2016; Accepted 15 November 2016

Copyright © 2016 Pablo Gil 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 presence of robotics and its application is becoming widespread. It is a reality that in the past few years robots have been revolutionizing the manufacturing and production as indicated by the International Federation of Robotics (IFR) in World Robotics Statistics. But also, robots are being successfully used in other fields, outside industry. Therefore, service robots are conquering various areas such as the domestic environment, transport, agriculture, security, education, medical and health care, and personal or professional assistance, among others. In these contexts, the sensing techniques are essential to enable robots to perform tasks autonomously. The sensorial perception from visual/tactile sensors is vital to develop autonomous robots with abilities for specific applications. On the one hand, visual sensors are widely used in mobile robots for mapping and exploration by land, sea, or air. But also, visual sensors are useful to recognize and locate objects in an environment. On the other hand, tactile sensors provide data for intelligent manipulation of located objects and to add semantic information such as hardness, flexibility, elasticity, roughness, or shape. Consequently, both sensors working together or separately allow researchers to implement new approaches, methods, and algorithms to achieve a robust sensing of dynamic and complex environments, improving the robot's abilities for specific applications in which there is interaction between a robot and its surrounding area.

This special issue consists of eight articles on various topics about robotics perception from six countries: China, Croatia, Germany, Korea, Mexico, and Spain. The aim of this special issue is an attempt to gather and cover recent

advances in robotic perception. In particular, the editors wish to explore the challenges and solutions to improve robot perception in both indoor and outdoor environments.

The location and navigation problem has traditionally been addressed for land mobile robots in indoors and more recently for robots and autonomous vehicles in outdoors. In this line of work, the paper by J. Duque et al. presents a new indoor positioning system based on the combination of data obtained from a Wi-Fi signal and RGB-D images acquired from cameras based on Time of Flight technology to estimate people location in indoor environments. The proposed system is able to detect more than one person in the same room using a nonintrusive method and low cost and easy installation technology. Besides, the paper by L. Fernández et al. presents a comparison among five known image descriptors such as SURF with Harris corner detector, HOG, DFT, Fourier signature, and gist-Gabor descriptor. The authors assess these descriptors using spherical panorama images obtained from the services of Google Street View. The goal is to use the proposed descriptors to build outdoor visual maps from Google images which are applied to both autonomous navigation and localization processes of mobile robots. The descriptors goodness to achieve that goal is measured using the relationship between precision and recall for each descriptor as well as the computational cost.

Recently, the research which proposes solutions for navigation problems has been applied to other contexts, resulting in marine robots and underwater vehicles. In this field, F. Mandic et al. propose a method for navigation by tracking an underwater target with a robot marine named BUDDY

AUV which uses data fusion between USBL measurements and sonar images in real scenarios. The proposed algorithm obtains precise and reliable underwater object tracking at steady rate, even in cases when either sonar or USBL measurements are faulty or are not available. Moreover, the paper by R. Pérez-Alcocer et al. addresses the underwater navigation problem in poor visibility conditions. The underwater image tends to be blurred and/or colour depleted. The authors have developed a visual system based on  $\alpha\beta$  space colour for detection of artificial landmarks underwater in poor visibility conditions without requiring the adjustment of parameters when marine environmental conditions are changed. This visual system has been integrated in a navigation control system. Furthermore, in the same line as these aforementioned works, A. Maldonado-Ramírez and L. A. Torres-Méndez present a method of detection and tracking of visual targets which can be relevant for ocean bed exploration. Authors have demonstrated the method's effectiveness with experiments applied to explorations of natural underwater structures like coral reefs carried out by a marine robot guided by visual features but with no human intervention.

Service robots tend to be autonomous robots with intelligence to perform behaviours in the real world. Probabilistic methods and algorithms are a growing area in the field of robotics. Accordingly, probabilistic robotics is widely used to estimate the robot pose and location as well as planning and controlling their trajectory and movements. Probabilistic robotics uses statistics and mathematical tools of artificial intelligence such as Bayes/Kalman/Particle filters as well as other Markov and Monte Carlo techniques. In this way, the paper by C. Rink et al. is focused on Monte Carlo registration methods. They present techniques for object modelling and pose estimations for further manipulation by a robot. Authors show various experiments with depth images acquired from Time-of-Flight camera and laser striper in real-time.

J.-H. Kim et al. show a detection method based on visual feature selection for autonomous firefighting robot. In that work, authors use FLIR cameras to classify fire, smoke, and both thermal reflections in indoor fire environments where there is dense smoke with bad visibility. The cameras were mounted in SAFFiR robot to extract motion information and texture features. Additionally, a Bayesian classification is carried out to probabilistically identify multiple instances of each target in real-time.

Robotic applications as intelligent manipulation often require not only sense of sight but also sense of touch. In these cases, the robots mount grippers and hands at the effector which is equipped with a tactile sensing system. Design, performance, and fabrication of tactile systems are usually based on a conductive material and/or a circuit of networked resistive sensor arrays. The wire features and the connections among electronic components cause problems such as crosstalk. J. Wu and L. Wang's paper introduces the design of a new S-NSDE-EP circuit using two wires for every driving-electrode and every sampling-electrode to reduce the crosstalk caused by the connected cables in the 2D networked resistive sensor array.

## Acknowledgments

We wish to give thanks to the reviewers for their help in selecting papers for this special issue. Also, we would like to acknowledge all members of Editorial Board of Journal of Sensor for approving the publication of this special issue. Finally, the authors are grateful for both the efforts in the preparation of the manuscripts and the choice of this journal's special issue to publish their scientific and technical contributions.

*Pablo Gil  
Youssef Mezouar  
Markus Vincze  
Juan A. Corrales*



## Research Article

# A Study of Visual Descriptors for Outdoor Navigation Using Google Street View Images

**L. Fernández, L. Payá, O. Reinoso, L. M. Jiménez, and M. Ballesta**

*Department of Systems Engineering and Automation, Miguel Hernandez University, Avda. de la Universidad s/n, Elche, 03202 Alicante, Spain*

Correspondence should be addressed to L. Payá; [lpaya@umh.es](mailto:lpaya@umh.es)

Received 22 March 2016; Revised 24 August 2016; Accepted 2 November 2016

Academic Editor: Jose R. Martinez-de Dios

Copyright © 2016 L. Fernández et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A comparative analysis between several methods to describe outdoor panoramic images is presented. The main objective consists in studying the performance of these methods in the localization process of a mobile robot (vehicle) in an outdoor environment, when a visual map that contains images acquired from different positions of the environment is available. With this aim, we make use of the database provided by Google Street View, which contains spherical panoramic images captured in urban environments and their GPS position. The main benefit of using these images resides in the fact that it permits testing any novel localization algorithm in countless outdoor environments anywhere in the world and under realistic capture conditions. The main contribution of this work consists in performing a comparative evaluation of different methods to describe images to solve the localization problem in an outdoor dense map using only visual information. We have tested our algorithms using several sets of panoramic images captured in different outdoor environments. The results obtained in the work can be useful to select an appropriate description method for visual navigation tasks in outdoor environments using the Google Street View database and taking into consideration both the accuracy in localization and the computational efficiency of the algorithm.

## 1. Introduction

Designing vehicles capable of navigating autonomously, in a previously unknown environment and with no human intervention, is a fundamental objective in mobile robotics. To achieve this objective, the vehicle must be able to build a model (or map) of the environment and to estimate its position within this model. A great variety of localization approaches can be found in the literature. In general, the position and orientation of the robot can be obtained from proprioceptive (odometer) or exteroceptive (laser, camera, or sonar) sensors, as presented in the works of Thrun et al. [1] and Gil et al. [2].

With the exteroceptive approach, the use of computer vision to create a representation of the environment is very extended due to the good relationship *quantity of information/cost* that the cameras offer. The research developed during the last years in the topic of map creation using visual information is enormous, and new algorithms are presented continuously. Usually, one of the key points of these

algorithms is the description of the visual information to extract relevant information which is useful for the robot to estimate its position and orientation. In general, the problem can be approached from two points of view: local features extraction and global-appearance approaches. In the first one, a number of landmarks (distinctive points or regions) are extracted from each scene and each landmark is described to obtain a descriptor which is invariant against changes in the robot position and orientation. Murillo et al. [3] presented an algorithm that made use of the SURF (Speeded Up Robust Features) description method [4] to improve the performance of appearance-based localization methods using omnidirectional images in large data sets. On the other hand, global-appearance approaches consist in representing each scene by a single descriptor which is computed working with the scene as a whole, with no local feature extraction. This approach has recently become popular and some examples can be found. Rossi et al. [5] present a metric to compute the image similarity using the Fourier Transform of spherical omnidirectional images in order to carry out the localization

of a mobile robot. Payá et al. [6] present a framework to carry out multirobot route following using an appearance-based approach with omnidirectional images to represent the environment and a probabilistic method to estimate the localization of the robot. Finally, Fernández et al. [7] deal with the problem of robot localization using the visual information provided by a single omnidirectional camera mounted on the robot, using techniques based on the global appearance of panoramic images and a Monte Carlo Localization (MCL) algorithm [8].

The availability of spherical images that represent outdoor environments is nowadays almost unlimited, thanks to the services of Google Street View. Furthermore, these images provide a complete 360-degree view of the scenery in the ground plane and 180-degree view vertically. Thanks to this great amount of information, these images can be used to carry out autonomous navigation tasks robustly. Using a set of these previously available spherical images as a dense visual map of an environment, it is possible to develop an autonomous localization and navigation system employing the images captured by a mobile robot or vehicle and comparing them with the map information in order to resolve the localization problem. This way, in this paper, we consider the use of the images provided by Google Street View as a visual map of the environment in which a mobile robot must be localized using the image acquired from an unknown position.

The literature regarding the navigation problem using Google Street View information is somewhat sparse but growing in recent years. For example, Gamallo et al. [9] proposed the combination of a low cost GPS with a particle filter to implement a vision based localization system that compares traversable regions detected by a camera with regions previously labeled in a map (composed of Google Maps images). The main contribution of this work is that a synthetic image of what the robot should see from the predicted position is generated and compared with the real observation to calculate the weight of each particle. Torii et al. [10] tried to predict the GPS location of a query image given the Google Street View database. This work presents a design of a matching procedure that considers linear combinations of bag-of-feature vectors of database images. With respect to indoor pose estimation, Aly and Bouguet [11] present an algorithm that takes as input spherical Google Street View images and as output their relative pose up to a global scale. Finally, Taneja et al. [12] proposed a method to refine the calibration of the Google Street View images leveraging cadastral 3D information.

The localization of the vehicle/robot can be formulated as the problem of matching the currently captured image with the images previously stored in the dense map (images in the database). Nowadays, a great variety of detection and description methods have been proposed in the context of visual navigation but, in our opinion, there exists no consensus on this matter when we use outdoor images.

Amorós et al. [13] carried out a review and comparison of different global-appearance methods to create descriptors of panoramic scenes in order to extract the most relevant information. The authors of this work developed a set of

experiments with panoramic images captured in indoor environments to demonstrate the applicability of some appearance descriptors to robotic navigation tasks and to measure their quality in position and orientation estimation. However, as far as outdoor scenarios are concerned, there is no revision of methods that offer good results. This situation, combined with the fact that using Google Street View images has barely been tested in autonomous navigation systems, has motivated the work presented here. Following this philosophy, we made a comparison between different descriptors of panoramic images but, in this case, we used Google Street View images captured in outdoor environments. This is a more challenging problem due to several features: the openness of the images (i.e., the degree of dominance of some structures such as the sky and the road which do not add distinctiveness to the image), their changing lighting conditions, and the large geometrical distance between the points where the images were captured.

Taking these features into account, we consider that it is worth carrying out a comparative evaluation of the performance of different image descriptors under real conditions of autonomous outdoor localization, since it would be a necessary step prior to the implementation of a visual navigation framework. In this paper, we evaluate two different approaches: approaches based on local features and approaches based on global appearance. In both cases we test the performance of the descriptor depending on the main parameters that configure it and we make a graphical representation of the *precision* of each method versus the *recall* [14].

When a robot has to navigate autonomously outdoors, very often a rough estimation of the area where the robot moves is available, and the robot must be able to estimate its position in this wide zone. This work focus on this task; we assume the zone where the robot navigates is approximately known and it must estimate its position more accurately in this area. With this aim, two different wide areas have been chosen to evaluate the performance of the localization algorithms, and a set of images per area has been obtained from the Google Street View database.

The remainder of the paper is organized as follows. In Section 2, we present the description methods evaluated in this work. In Section 3, the experimental setup and the databases we have used are described. Section 4 describes the method we have followed to evaluate the descriptors in a localization process. Section 5 presents the experimental results. Finally, in Section 6, we outline the conclusions and the future works.

## 2. Image Descriptors

In this section, we present five different image descriptors that are suitable to build a compact description of the appearance of each scene [13–15]. One of the methods, previously denoted as a feature-based approach, consists in representing the image as a set of landmarks extracted from the scene along with the description of such landmarks. The method selected for this landmarks description is SURF (Speeded Up Robust Features). The other methods chosen to carry out the comparative analysis are the following appearance-based methods: the two-dimensional Discrete Fourier Transform (DFT), the

Fourier Signature (FS), *gist*, and the Histogram of Oriented Gradients (HOG). Each method uses a different mechanism to express the global information of the scene. First, DFT and FS are based on the analysis in the frequency domain in two dimensions and one dimension, respectively. Second, the approach of *gist* we use is built from edges information, obtained through Gabor filtering and analyzed in several scales. Finally, HOG gathers systematic information from the orientation of the gradient in local areas of the image. The choice of these description methods will permit analyzing the influence of each kind of information in the localization process.

The initial objective of this study was to compare some global-appearance methods. However, we have decided to include in this comparative evaluation a local features description method to make a more complete study. With this aim, we have chosen SURF due to its relatively low computational cost comparing with other classical feature-based approaches.

The next subsections present briefly the description methods included in the comparative evaluation.

**2.1. SURF and Harris Corner Detector.** The Speeded Up Robust Features (SURF) were introduced by Bay et al. [4]. This study showed that SURF outperform existing methods with respect to repeatability, robustness, and distinctiveness of the descriptors. The detection method uses integral images to reduce the computational time and is based on the Hessian matrix. On the other hand, the descriptor represents a distribution of Haar-wavelet responses within the interest point neighborhood and makes an efficient use of integral images. In this work we only include the standard SURF descriptor, which has a dimension of 64 components per landmark, but there are two more versions: the extended version (E-SURF) with 128 elements and the upright version (U-SURF), that is not invariant to rotation and has a length of 64 elements. On the other hand, we perform the detection of the features using the Harris corner detector (based on the eigenvalues of the second moment matrix [16]) because our experiments showed that this method extracted most robust points in outdoor images comparing to the SURF extraction method.

This way the method we use in this work is a combination of these two algorithms. More specifically, the Harris corner detector is used to extract the features from the image, and the standard SURF descriptor is used to characterize and describe each one of the landmarks previously detected.

**2.2. Two-Dimensional Discrete Fourier Transform.** From an image  $f(x, y)$  with  $N_x$  rows and  $N_y$  columns, the 2D Discrete Fourier Transform (DFT) can be defined as follows:

$$\begin{aligned} F[f(x, y)] &= F(u, v) \\ &= \frac{1}{N_y N_x} \sum_{x=0}^{N_x-1} \sum_{y=0}^{N_y-1} f(x, y) \cdot e^{-2\pi j(ux/N_x + vy/N_y)}, \quad (1) \\ u &= 0, \dots, N_x - 1, \quad v = 0, \dots, N_y - 1, \end{aligned}$$

where  $(u, v)$  are the frequency variables and the transformed function  $F(u, v)$  is a complex function which can be decomposed into a magnitudes matrix and an arguments matrix. This transformation presents some interesting properties which are helpful in robot localization tasks. First, the most relevant information in the Fourier domain concentrates in the low frequency components, so it is possible to reduce the amount of memory and to optimize the computational cost by retaining only the first  $k_x$  rows and  $k_y$  columns in the transform. Second, when  $f(x, y)$  is a panoramic scene, a translation in the rows and/or columns of the original image produces a change only in the arguments matrix [15]. This way, the magnitudes matrix contains information which is invariant to rotations of the robot in the ground plane, and the arguments matrix contains information that can be useful to estimate the orientation of the robot in this plane with respect to a reference image (using the DFT shift theorem).

Taking these facts into account, the global description of the image  $f(x, y)$  consists of the magnitudes matrix  $A(u, v)$  and the arguments matrix  $\Phi(u, v)$  of its two-dimensional DFT. The dimensions of both matrices are  $k_x < N_x$  rows and  $k_y < N_y$  columns. On the one hand,  $A(u, v)$  is useful to estimate the robot position and, on the other hand, the information in  $\Phi(u, v)$  can be used to estimate the robot orientation.

**2.3. Fourier Signature.** The third image description method used in this comparative analysis is the Fourier Signature (FS), described initially by Menegatti et al. [17]. From an image  $f(x, y)$  with  $N_x$  rows and  $N_y$  columns, the FS consists in obtaining the one-dimensional DFT of each row. This method presents some advantages, such as its simplicity, its low computational cost, and the fact that it exploits better the invariance against rotations of the robot in the ground plane when we work with panoramic views.

More specifically, the process to compute the FS consists in transforming each row  $x$  of the original panoramic image  $\{f_x\} = \{f_{x,0}, f_{x,1}, \dots, f_{x,N_y-1}\}$ ,  $x = 0, \dots, N_x - 1$ , into the sequence of complex numbers  $\{F_x\} = \{F_{x,0}, F_{x,1}, \dots, F_{x,N_y-1}\}$ ,  $x = 0, \dots, N_x - 1$ , according to the 1D-DFT expression:

$$\begin{aligned} F_{x,k} &= \sum_{n=0}^{N_y-1} f_{x,n} \cdot e^{-j(2\pi/N_y)kn}, \quad (2) \\ k &= 0, \dots, N_y - 1, \quad x = 0, \dots, N_x - 1. \end{aligned}$$

The result is a complex matrix  $F(x, v)$ , where  $v$  is a frequency variable, which can be decomposed into a magnitudes matrix and an arguments matrix.

Thanks to the 1D-DFT properties it is possible to represent each row of  $F(x, v)$  with the first coefficients since the most relevant information is concentrated in the low frequency components of each row in the descriptor, so it is possible to reduce the amount of memory by retaining only  $k_y$  first columns in signature  $F(x, v)$ . Also, when  $f(x, y)$  is a panoramic scene, the modules matrix is invariant against robot rotations in the ground plane and the magnitudes matrix permits estimating the change in the robot orientation using the DFT shift theorem [15, 17, 18].



Taking these facts into account, the global description of the image  $f(x, y)$  consists of the magnitudes matrix  $A(x, v)$  and the arguments matrix  $\Phi(x, v)$  of the Fourier Signature. The dimensions of both matrices are  $N_x$  rows and  $k_y < N_y$  columns. First, the position of the robot can be estimated using the information in  $A(x, v)$ , since it is invariant to changes in robot orientation and second  $\Phi(x, v)$  can be used to estimate the robot orientation.

**2.4. Gist.** The concept of the *gist* of an image can be defined as an abstract representation that activates the memory of scene categories [19]. The *gist*-based descriptors try to represent the image by obtaining its essential information simulating the human perception system and its ability to recognize a scene through the identification of color saliency or remarkable structures. Torralba [20] presents a model to obtain global scene features, working in several spatial frequencies and using different scales based on Gabor filtering. They use these features in a scene recognition and classification task. In previous works [13] we employed a *gist*-Gabor descriptor in order to obtain frequency and orientation information. Due to the good results obtained in indoor environments when the mobile robot presents 3 DOF (degrees of freedom) movements on the ground plane, the fourth method employed in the comparative analysis presented in this paper is the *gist* descriptor of panoramic images.

The method starts with two versions of the initial panoramic image  $f(x, y)$ : the original one, with  $N_x$  rows and  $N_y$  columns, and a new version after applying a Gaussian low-pass filter and subsampling to a new size equal to  $0.5 \cdot N_x \times 0.5 \cdot N_y$ . After that, both images are filtered with a bank of  $n_f$  Gabor filters whose orientations are evenly distributed to cover the whole circle. Then, to reduce the amount of information, the pixels into both images are grouped into  $k_1$  horizontal blocks per image, whose width is equal to  $N_y$  in the first image and  $0.5 \cdot N_y$  in the second one. The average value of the pixels in each group is calculated and all this information is arranged into a final descriptor, which is a column vector  $\vec{g}$  with  $2 \cdot k_1 \cdot n_f$  components. This descriptor is invariant against rotations of the vehicle on the ground plane. More information about the method can be found in [13].

**2.5. Histogram of Oriented Gradients.** The Histogram of Oriented Gradients (HOG) descriptors are based on the orientation of the gradient in local areas of an image. It was described initially by Dalal and Triggs [21]. More concisely, it consists first in obtaining the magnitude and orientation of the gradient of each pixel of the original image. This image is divided then into a set of cells and a histogram of gradient orientation is compiled for each cell, aggregating the information of the gradient orientation of each pixel within the cell, weighting with the magnitude of the pixel.

The omnidirectional images captured from a specific position of the ground plane contain the same pixels in a row, independently on the orientation of the robot in this plane, but in a different order. Taking this fact into account, if we calculate the histogram of cells that have the same width of

the original image, we obtain a descriptor which is invariant against rotations of the robot.

The method we use is described in depth in [22] and can be summarized as follows. The initial panoramic image  $f(x, y)$  with  $N_x$  rows and  $N_y$  columns is first filtered to obtain two images with the information of the horizontal and vertical edges,  $f_x(x, y)$  and  $f_y(x, y)$ . From these two images, the magnitude of the gradient and its orientation is obtained, pixel by pixel, and the results are stored in matrices  $M(x, y)$  and  $\Theta(x, y)$ , respectively. Matrix  $\Theta(x, y)$  is then divided into  $k_2$  horizontal cells, whose width is equal to  $N_y$ . For each cell, an orientation histogram with  $b$  bins is compiled. During this process, each pixel in  $\Theta(x, y)$  is weighted with the magnitude of the corresponding pixel in  $M(x, y)$ . At the end of the process, the set of histograms constitutes the final descriptor  $\vec{h}$  which is a column vector with  $k_2 \cdot b$  components.

### 3. Experiments Setup

The main objective of this work consists in carrying out an exhaustive evaluation of the performance of the description methods presented in the previous section. All these methods will be included in a localization algorithm and their performance will be evaluated and compared both in terms of computational cost and localization accuracy. The results of this comparative evaluation will give us an idea of which is the description method that offers the best results in outdoor environments when using Google Street View images.

With this aim, two different regions in the city of Elche (Spain) have been selected and the Google Maps images of these two areas have been obtained and stored in two data sets. Each one of these data sets will constitute a map and will be used subsequently to estimate the position of the vehicle within the map by comparing the image captured by the vehicle from the unknown position with the images previously stored in each map.

The main features of the two sets of images are as follows.

**Set 1.** Set 1 consists of 177 full spherical panorama images with resolution generally up to  $3328 \times 1664$  pixels. Each image covers a field of view of 360 degrees in the ground plane and 180 degrees vertically. Figure 1 shows the GPS position where each image was captured (blue dots) and two examples of the panoramic images after a preprocessing process. This set corresponds with a mesh topography database that contains images of various streets and open areas. The images cover an area of approximately  $700 \text{ m} \times 300 \text{ m}$ .

**Set 2.** Set 2 consists of 144 full spherical panorama images. The images have been captured along the same street with a linear topology covering approximately 1700 m. The appearance of these images is more urban. Figure 2 shows the GPS position where each image was captured (blue dots) and three examples of the panoramic images after a preprocessing process.

**3.1. Image Preprocessing and Map Creation.** Due to the wide vertical field of view of the acquisition system, the sky is



FIGURE 1: Bird eye's view of the region chosen as map 1 prior to the localization experiment. The blue dots represent the coordinates where the images of the Set 1 were captured. Two examples of Google Street View images after a preprocessing step are also shown.



FIGURE 2: Bird eye's view of the region chosen as map 2 prior to the localization experiment. The blue dots represent the coordinates where the images of Set 2 were captured. Three examples of Google Street View images after a preprocessing step are also shown.

often a big portion of the Google Street View images. The appearance of this area will be very prone to changes when the localization process is carried out in a different time of day with respect to the time of day when the map was captured. Taking this fact into account, a preprocessing step has been carried out to remove part of the sky in the scenes.

Once part of the sky has been removed from all the scenes, the images are converted into grayscale and their resolution is reduced to  $512 \times 128$  pixels, to ensure the computational viability of the algorithms.

After that, each image will be described using the five description methods presented in Section 2. At the end, one map will be available per image set and per description

method. Each map will be composed of the set of descriptors of each panoramic scene.

**3.2. Localization Process.** Once the maps are available, in order to evaluate the different visual descriptors introduced in Section 2 to solve the localization problem, we also make use of Google Street View images.

To carry out the localization process, first we choose one of the images of the database (named as *test image*). In this moment, this image is removed from the map. Second, we compute the descriptor of the test image (using one of the methods presented in Section 2) and obtain the distance between this descriptor and the descriptors of the rest of



the images stored in the corresponding map. As a result, the images of the map are arranged from the nearest to the farthest using the image distance as arranging criterion.

The result of the localization algorithm is considered correct if the first image it returns was captured on the map point which is geometrically the closest to the test image capture point (the GPS coordinates are used with this aim). We will refer to this case as a correct localization in *zone 1*. However, since this is a quite challenging and restrictive problem, it is also interesting to know if the first image that the algorithm returns was captured on one of the two geometrically closest points to the test image capture point (*zone 2*) or even on one of the three geometrically closest points (*zone 3*). The first case is the ideal one, but we are also interested in the other cases as they will indicate if the algorithm is returning an image in the surroundings of the actual position of the test image (i.e., the localization algorithm detects that the robot is in a zone close around its actual position).

This process is repeated for each description method, using all the images of Sets 1 and 2 as test images. In brief, the procedure to test the localization methods previously explained consists in the following steps, for each image and description method:

- (1) Extracting one image of the set (denoted as test image); then, this test image is eliminated from the map
- (2) Calculating the descriptor of the test image
- (3) Calculating the distance between this descriptor and all the map descriptors, which we named *image distance*
- (4) Retaining the most similar descriptor and studying if it corresponds to one image that has been captured in the surroundings of the test image capture point (*zone 1*, *2*, or *3*)

As a result, the next data are retained for an individual test image: the *image distance* between the test image descriptor and the most similar map descriptor,  $D^t$ , and the localization results in *zone 1* (correct or wrong match),  $m_1^t$ , in *zone 2*,  $m_2^t$ , and in *zone 3*,  $m_3^t$ . After repeating this process with all the test images, the results will consist of four vectors, whose dimension is equal to the number of test images. The first vector,  $\vec{D}$ , contains the distances,  $D^t$ , and the other three,  $\vec{m}_1$ ,  $\vec{m}_2$ , and  $\vec{m}_3$ , contain, respectively, the information of correct or incorrect matches in *zones 1*, *2*, and *3*.

#### 4. Evaluation Methods

In this work, the localization results are expressed by means of *recall* and *precision* curves [14]. To build them, the components of the vectors  $\vec{D}$ ,  $\vec{m}_1$ ,  $\vec{m}_2$ , and  $\vec{m}_3$  are equally sorted in ascending order of the distances that appear in the first vector. The resulting sorted vectors of correct and wrong matches are then used to calculate the values of *recall* and *precision*. Let us focus on the sorted vector of matches in *zone 1*,  $\vec{m}_1^s$ . First, for each component in this vector, the *recall* is calculated as the number of correct matches obtained so far with respect to

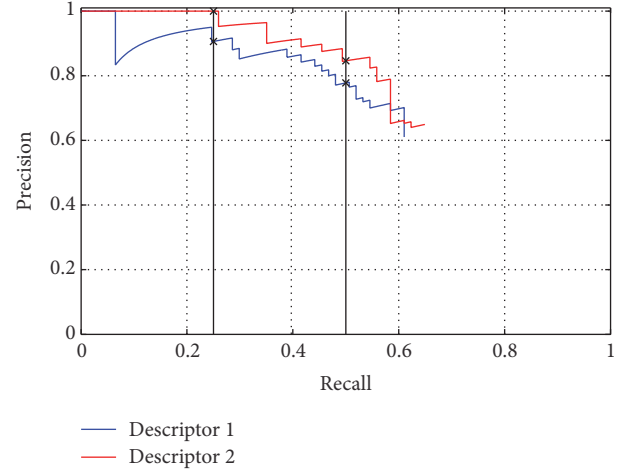


FIGURE 3: Two sample *recall-precision* graphs obtained after carrying out the localization experiments with two different description methods.

the total number of test images. Second, for each component in the same vector, the *precision* is obtained as the number of correct matches obtained so far with respect to the number of test images considered so far. Then, with the information contained in these vectors, a *precision versus recall* curve is built, corresponding to the localization in *zone 1*. This is repeated with the sorted vectors  $\vec{m}_2^s$  and  $\vec{m}_3^s$  to obtain the localization results in *zone 2* and *zone 3*.

In our experiments, the most important piece of information of this type of graphs is the final point because it shows the global result of the experiment (final precision after considering all the test images). However, additional relevant information can be extracted from them, because the graph also shows the ability of the localization algorithm to find the correct match while considering a specific *image distance* threshold. As explained in the previous paragraph, the results have been arranged considering the ascending value of distances. Taking it into account, as the recall increases, the threshold also does. For this reason, the evolution of the *recall-precision* curves contains information about the robustness of the algorithm with respect to a specific *image distance* threshold. If the *precision* values stay high, independently of the *recall*, this shows the presence of a lower number of wrong results under this distance threshold. Figure 3 shows two sample *recall-precision* curves obtained after running the localization algorithm with all the test images and two different description methods, considering *zone 1*. Both curves show a similar final *precision* value, between 0.6 and 0.65. However, the evolutions present a different behavior. As an example, if we consider as threshold the distance associated to *recall* = 0.25, according to the graph, the precision of descriptor 1 is 100%, but the precision of descriptor 2 is 90%. This means that, considering the selected *image distance* threshold, 25% of correct localizations are achieved with 100% of probability using descriptor 1 and with a 90% of probability using descriptor 2. This study can be carried out considering any value for the *image distance* threshold.



Before running the algorithm, it is necessary to define the *image distance*. We use two different distance measures depending on the kind of descriptor used.

First, in the case of the feature-based method (SURF-Harris), it is necessary to extract the interest points prior to describing the appearance of the image. We propose using the Harris corner detector [16] to extract visual landmarks from the panoramic images. After that, each interest point is described using standard SURF. To compare the test image with the map images, first we extract and describe the interest points from all the images. After that, a matching process is carried out with these points. The points detected in the test image, captured with a particular position and orientation of the vehicle, are searched in the map images. The performance of the matching method is not the scope of this paper; we only employ it as a tool. Once the matching process has been carried out, we evaluate the performance of the descriptor taking into account the number of matched points, so that we will consider as closest image the one that presents more matching points with the test image. More concisely, we compute the distance between the test image  $t$  and any other image of the map  $j$  as

$$D_{\text{fea}}^{tj} = 1 - \left( \frac{NM^{tj}}{\max_j (\overrightarrow{NM^t})} \right), \quad (3)$$

where  $NM^{tj}$  is the number of matches between the images  $t$  and  $j$ ,  $\overrightarrow{NM^t} = [NM^{t1}, \dots, NM^{tn_{\text{map}}}]$  is a vector that contains the number of matches between the image  $t$  and every image of the map, and  $n_{\text{map}}$  is the number of images in the map.

Second, in the case of appearance-based methods (2D DFT, FS, *gist*, and HOG), no local information needs to be extracted from the images. Instead, the appearance of the whole images is compared. This way, the global descriptor of the test image is calculated and the distances between it and the descriptors of the map images are obtained. The Euclidean distance is used in this case, defined as

$$D_E^{tj} = \sqrt{\sum_{m=1}^M (\vec{d}_t^m - \vec{d}_j^m)^2}, \quad (4)$$

where  $\vec{d}_t$  is the descriptor of the test image  $t$ ,  $\vec{d}_j$  is the descriptor of the map image  $j$ , and  $M$  is the size of the descriptors. This distance is normalized to obtain the final distance between the images  $t$  and  $j$ , according to the next expression:

$$D_{\text{app}}^{tj} = \frac{D_E^{tj}}{\max_j (\vec{D}_E^t)}, \quad (5)$$

where  $D_E^{tj}$  is the Euclidean distance between the descriptor of the test image  $t$  and the map image  $j$ ,  $\vec{D}_E^t = [D_E^{t1}, \dots, D_E^{tn_{\text{map}}}]$  is a vector that contains the Euclidean distance between the descriptor of the image  $t$  and all the images in the map, and  $n_{\text{map}}$  is the number of images in the map.

It is important to note that the algorithm must be able to estimate the position of the robot with accuracy, but it is also important that the computational cost is adequate, to know whether it would be feasible to solve the problem in real time. To estimate the computational cost, we have computed, considering both maps in the experiments, the necessary time to calculate the descriptor of each test image, to compute the distance to the map descriptors and to detect the most similar descriptor. We must take into account that the descriptors of all the map images can be computed prior to the localization, in an off-line process. Apart from the time, we have also estimated the amount of memory needed to store each image descriptor.

To finish, we also propose to study the relationship *distance between two image descriptors* versus *geometric distance between the capture points of these two images*. Ideally, the distance between the descriptors must increase as the geometric distance between capture points does (i.e., it must not present any local minima). This information is very interesting in applications such as map building, where the robot must be able to build a map using as input information only the distance between image descriptors. It is also important when it is necessary to estimate the position of the vehicle at halfway points within the grid map. Additionally, it may help to detect if the problem of *visual aliasing* is present in the environment (i.e., two zones which are geometrically far may present a similar visual appearance, which might lead to errors in the mapping and localization process).

## 5. Experimental Results

As stated in the previous section, with the purpose of establishing the capacity of each descriptor to correctly localize the robot (or vehicle), we have built *recall-precision* curves to reflect the results of each experiment. Figure 4 shows this graphical representation using (a) the first and (b) the second set of images (denoted as Sets 1 and 2 in previous sections). To build this figure, we consider the localization results in *zone 1*. This way, the figure shows the ability of the localization algorithm to correctly detect which image of the map was captured closer to the test image. This is the most restrictive case.

Apart from it, the performance of the localization algorithm in *zones 2* and *3* has also been studied. This way, Figure 5 shows the results of the localization process in *zone 2* using (a) Set 1 and (b) Set 2. Finally, Figure 6 shows the localization results in *zone 3* using (a) Set 1 and (b) Set 2. This is the least restrictive case among the three studied.

In all cases, the results show that the SURF-Harris descriptor presents a relatively better performance comparing to the other descriptors, in terms of accuracy and using both image sets. As far as the methods based on global appearance are concerned, the good behavior of HOG can be highlighted. In the case of the localization in *zone 2* it reaches 60% and 50% of precision in Sets 1 and 2, respectively. These results can be considered relatively good, taking into account the fact that the localization process is solved in an absolute way (i.e., we consider that no information about the previous position of the robot is available and the test image is compared with all the images stored in the data sets). In a real application, it

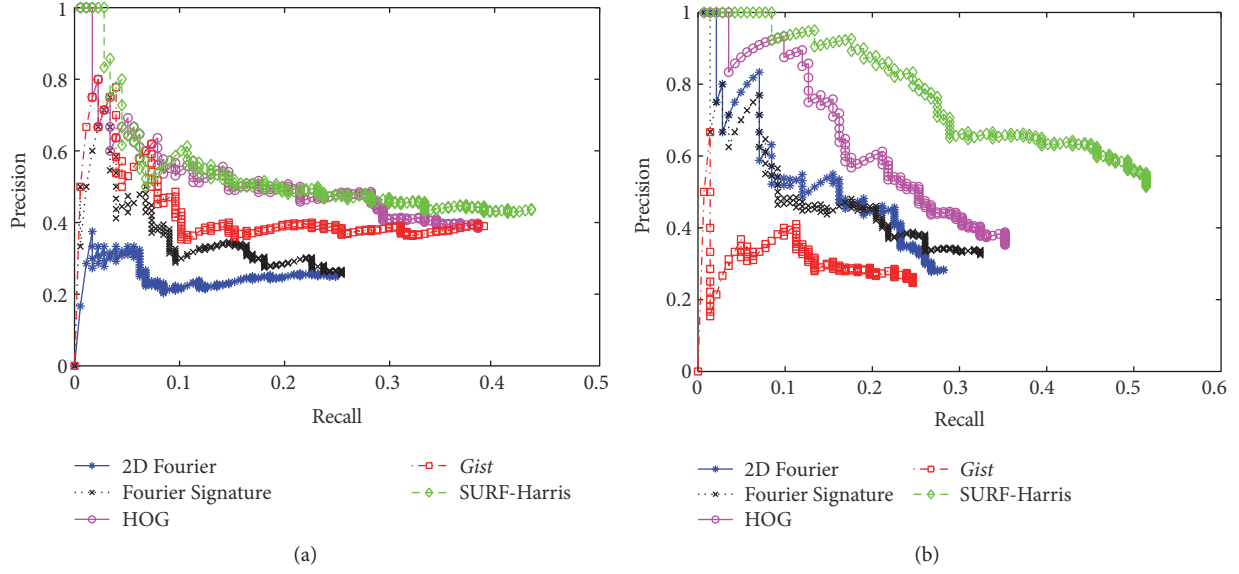


FIGURE 4: Results of the localization algorithm considering the correct matches in *zone 1* using (a) Set 1 and (b) Set 2. The results of each description method are shown as different recall-precision curves.

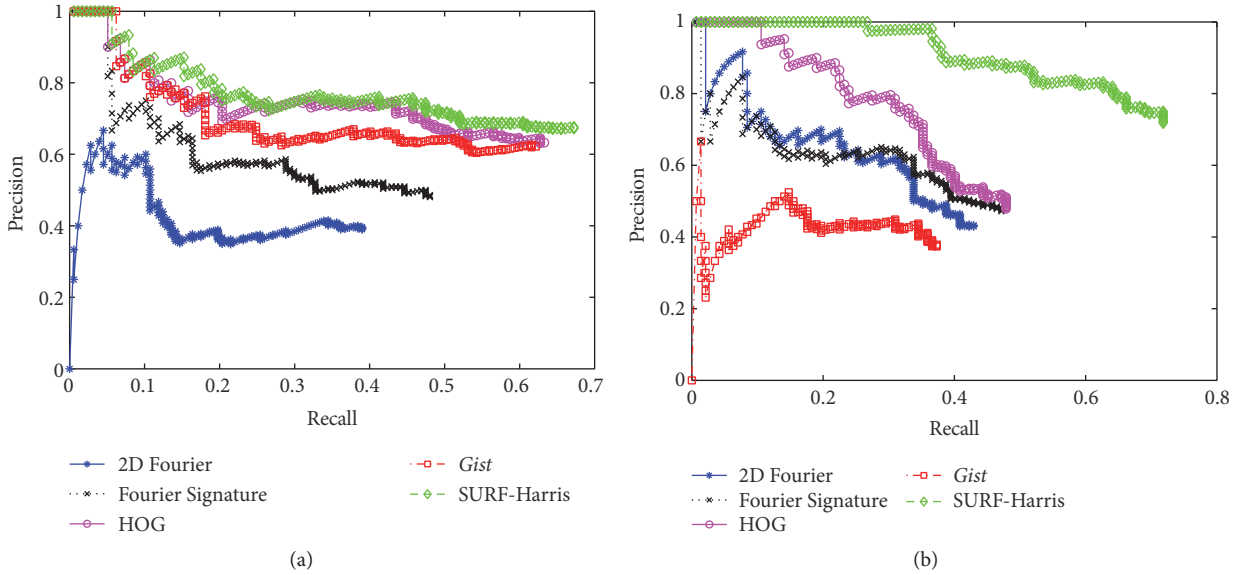


FIGURE 5: Results of the localization algorithm considering the correct matches in *zone 2* using (a) Set 1 and (b) Set 2. The results of each description method are shown as different recall-precision curves.

is usual to make use of any kind of probabilistic algorithm to estimate the position of the robot taking into account its previous estimated position. This is expected to provide a higher accuracy. We expect to develop this type of algorithms and tests in a future work.

Some additional conclusions can be reached by comparing the performance of the methods in open areas (Set 1) and urban areas (Set 2). In open areas, the performance of SURF-Harris, HOG, and *gist* is quite similar and relatively good in all cases, and the methods based on the Discrete Fourier Transform tend to present worse results. However, in the case

of urban areas, SURF-Harris outperforms the other methods, and *gist* is the one that presents the worst results.

Apart from the localization accuracy, it is also important to study the computational cost of the process, since in a real application it would have to run in real time, as the robot is navigating through the environment. This way, we have obtained in all cases the necessary time to calculate the descriptor of the test image on the one hand and to compare it with the descriptors stored in the map, to detect the most similar descriptor and to analyze the results on the other hand. The average computational time of the localization process

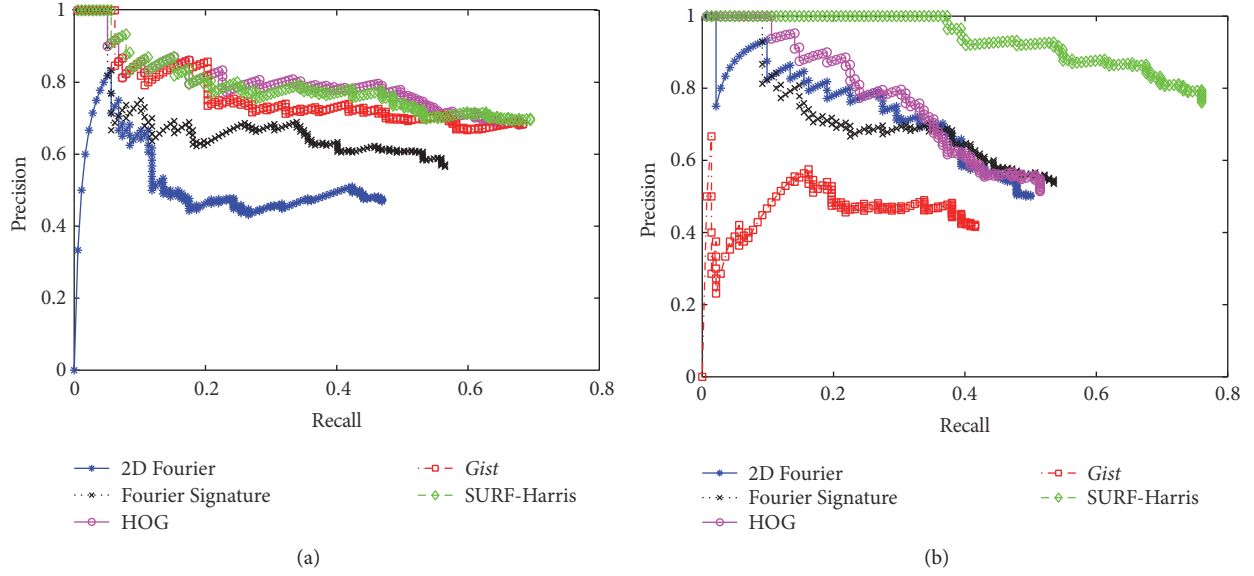


FIGURE 6: Results of the localization algorithm considering the correct matches in *zone 3* using (a) Set 1 and (b) Set 2. The results of each description method are shown as different recall-precision curves.

TABLE 1: Average computational cost of the description algorithms studied, per test image. For each description method and data set, the table shows first the necessary time to obtain the test image descriptor and second the time to compare it with the descriptors of the map and to obtain the final localization result.

	2D Fourier	Fourier Signature	<i>Gist</i>	HOG	SURF-Harris
Data Set 1 Descriptor	0.0087 s	0.0080 s	0.4886 s	0.0608 s	0.5542 s
Data Set 1 Match	0.0015 s	0.0058 s	0.0006 s	0.0008 s	25.8085 s
Data Set 2 Descriptor	0.0085 s	0.0079 s	0.4828 s	0.0621 s	0.5389 s
Data Set 2 Match	0.0012 s	0.0047 s	0.0005 s	0.0006 s	19.3931 s

after considering all the test images is shown in Table 1. To obtain the results of this table, the algorithms have been implemented using Matlab.

With respect to the computational cost, the methods based on the Fourier Transform are significantly faster than the rest, while SURF-Harris presents a considerably high computational cost. About the necessary time to compare two descriptors, *gist* and HOG are the fastest methods. In the case of SURF-Harris, the brute force match method implemented results in a relatively high computational cost. This method has been chosen to make a homogeneous comparison with the other global-appearance methods. However, in a real implementation, a bag-of-words based approach [23] would improve the computational efficiency of the algorithm.

At last, we have obtained the average memory size needed to store each descriptor. The results are shown in Table 2. *Gist* is the most compact descriptor (it is able to compress the information in each scene significantly) while SURF-Harris needs more memory size.

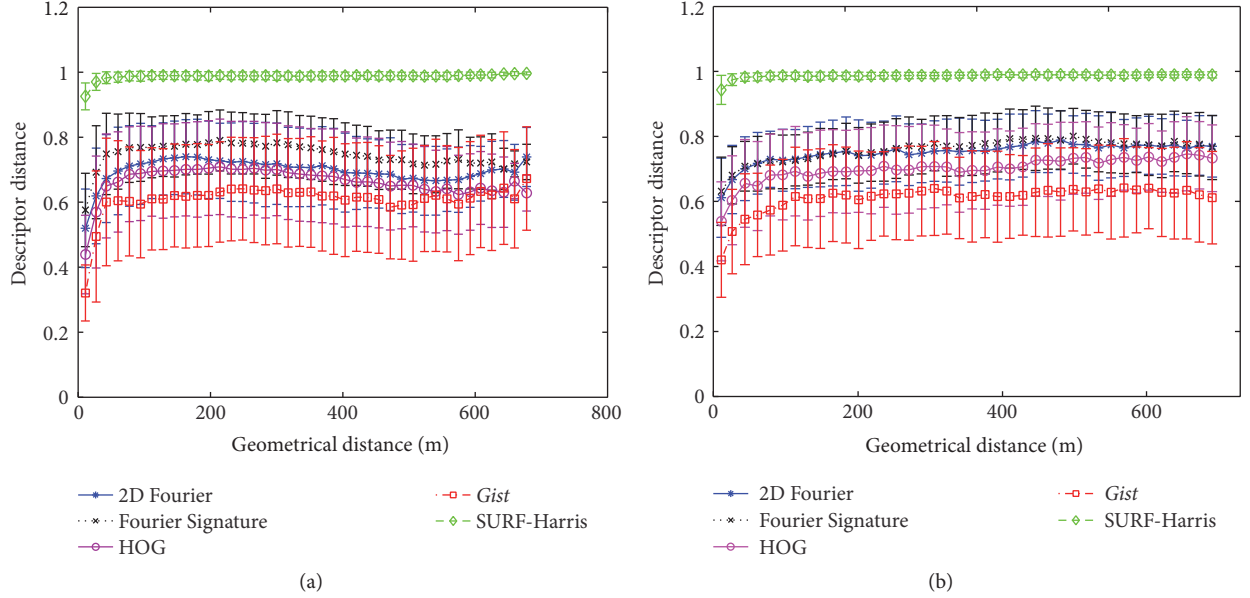
Considering these results jointly with the precision in localization, we could say that the SURF-Harris descriptor shows very good results in location accuracy but its computational cost makes it unfeasible to solve a real application. HOG, which is the second in terms of accuracy, also has a very good computational cost, so we consider it interesting to

study more thoroughly this descriptor as future work and to implement more advanced versions of this method to try to optimize the accuracy. Likewise, other types of distances to compare images could also be studied, apart from the Euclidean distance. For the same reasons, we also consider it appropriate to examine more thoroughly the *gist* descriptors, as well as using other methods to extract the *gist* of a scene apart from the orientation information (e.g., from the color information).

As a final experiment, we have studied the relationship *distance between two image descriptors versus geometric distance between the capture points of these two images*. As stated at the beginning of the section, this information is very interesting in some applications such as the construction of maps from the images, with geometric precision, or the localization of the vehicle at halfway points of the grid of the map. It is important that the distance between descriptors grows as the geometric distance does. Figure 7 shows the results obtained using (a) Set 1 and (b) Set 2. To obtain these figures, an image has been set as a reference image, and the distance between the reference image descriptor and the other descriptors has been calculated. The figure shows this distance versus the geometric distance between the capture points of each image and the capture point of the reference image. In both cases, this relationship is monotonically increasing up to a geometric

TABLE 2: Necessary memory to store each descriptor.

	2D Fourier	Fourier Signature	Gist	HOG	SURF-Harris
Descriptor	16384 bytes	32768 bytes	4096 bytes	8192 bytes	110400 bytes

FIGURE 7: Relationship *distance between two image descriptors* versus *geometric distance between the capture points of these two images* using (a) Set 1 and (b) Set 2.

distance of approximately 100 meters. From this point it tends to stabilize with a relatively high variance. The exception is the local features descriptor, which stabilizes at the final value from a very small geometrical distance. However, the appearance-based descriptors exhibit a more linear behavior around each image.

## 6. Conclusions and Future Works

In this paper, we have carried out a comparative evaluation of several description methods of scenes, considering the performance of these methods to accurately solve an absolute localization problem in a large real outdoor environment. We evaluated two different approaches of visual descriptors, local features descriptors (SURF-Harris), and global-appearance descriptors (2D Discrete Fourier Transform, Fourier Signature, HOG, and *gist*).

All the tests have been carried out with images of Google Street View, captured under realistic conditions. Two different areas of a city have been considered, an open area and a purely urban area with narrower streets. The capture points of each area present different topography. The first one is a grid map that covers several streets and avenues and the second one a linear map (i.e., the images were captured when the mobile traversed a linear path on a narrow street).

Some different studies have been performed. First, we have evaluated the accuracy of the localization process. To

do this, *recall* and *precision* curves have been computed to compare the performance of each description method. We plot the *recall* and *precision* curves for both areas, taking into account different levels of accuracy to consider that the localization result is correct. In these experiments, the computational cost of the localization process has also been analyzed.

We have also studied each descriptor in terms of behavior of the descriptor distance comparing to geometrical distance between image capture points. To do this, we plot a curve that represents the descriptor distance versus the geometrical distance between capture points. This measure is very useful for performing navigation tasks, since thanks to it we can estimate the range of use of the descriptor.

It is noticeable that the SURF-Harris descriptor is the most suitable descriptor in terms of precision in localization, but it presents a smaller zone of work in terms of Euclidean distance between descriptors. The HOG descriptor has shown a relatively good performance to solve the localization problem and presents a good response of the descriptors distance versus geometrical distance between capture points. If we analyze jointly the results of both experiments and take into account the computational cost (Tables 1 and 2) we conclude that, although the SURF-Harris descriptor presents the best results in terms of recall and precision curves, it does not allow us to work in real time. Therefore, taking into account



that HOG is the descriptor that presents the second best results in terms of recall and precision curves and allows us to work in real time, we can conclude that the HOG is the most suitable descriptor.

We plan to extend this work to (a) capture a real outdoor trajectory traveling along several streets and capturing omnidirectional images using a catadioptric vision system, (b) combine the information provided by this vision system and the images of the Google Street View, and (c) evaluate the performance of the best descriptors in a probabilistic localization process.

## Competing Interests

The authors declare that they have no competing interests.

## 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 Projects AICO/2015/021, *Localización y Creación de Mapas Visuales para Navegación de Robots con 6 GDL*, and GV/2015/031, *Creación de Mapas Topológicos a Partir de la Apariencia Global de un Conjunto de Escenas*.

## References

- [1] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2001.
- [2] A. Gil, O. Reinoso, M. A. Vicente, C. Fernández, and L. Payá, "Monte carlo localization using SIFT features," in *Pattern Recognition and Image Analysis*, vol. 3522 of *Lecture Notes in Computer Science*, pp. 623–630, 2005.
- [3] A. Murillo, J. Guerrero, and C. Sagüés, "Surf features for efficient robot localization with omnidirectional images," in *Proceedings of the IEEE International Conference on Robotics & Automation*, San Diego, Calif, USA, 2007.
- [4] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: speeded up robust features," in *Proceedings of the 9th European Conference on Computer Vision*, Graz, Austria, 2006.
- [5] F. Rossi, A. Ranganathan, F. Dellaert, and E. Menegatti, "Toward topological localization with spherical fourier transform and uncalibrated camera," in *Proceedings of the International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN '08)*, pp. 319–330, Venice, Italy, 2008.
- [6] L. Payá, O. Reinoso, F. Amoros, L. Fernández, and A. Gil, "Probabilistic map building, localization and navigation of a team of mobile robots. application to route following," in *Multi-Robot Systems: Trends and Development*, pp. 191–210, 2011.
- [7] L. Fernández, L. Payá, D. Valiente, A. Gil, and O. Reinoso, "Monte Carlo localization using the global appearance of omnidirectional images: algorithm optimization to large indoor environments," in *Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics (ICINCO '12)*, pp. 439–442, Rome, Italy, July 2012.
- [8] M. Montemerlo, *FastSLAM: a factored solution to the simultaneous localization and mapping problem with unknown data association [Ph.D. thesis]*, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pa, USA, 2003.
- [9] C. Gamallo, P. Quintía, R. Iglesias-Rodríguez, J. V. Lorenzo, and C. V. Regueiro, "Combination of a low cost GPS with visual localization based on a previous map for outdoor navigation," in *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA '11)*, pp. 1146–1151, Cordoba, Spain, November 2011.
- [10] A. Torii, J. Sivic, and T. Pajdla, "Visual localization by linear combination of image descriptors," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops '11)*, pp. 102–109, Barcelona, Spain, November 2011.
- [11] M. Aly and J.-Y. Bouguet, "Street view goes indoors: automatic pose estimation from uncalibrated unordered spherical panoramas," in *Proceedings of the IEEE Workshop on the Applications of Computer Vision (WACV '12)*, pp. 1–8, Breckenridge, Colo, USA, January 2012.
- [12] A. Taneja, L. Ballan, and M. Pollefeys, "Registration of spherical panoramic images with cadastral 3d models," in *Proceedings of the International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT '12)*, pp. 479–486, Zurich, Switzerland, 2012.
- [13] F. Amorós, L. Payá, O. Reinoso, and L. Jiménez, "Comparison of global-appearance techniques applied to visual map building and localization," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, pp. 395–398, Rome, Italy, 2012.
- [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] L. Payá, L. Fernandez, O. Reinoso, A. Gil, and D. Ubeda, "Appearance-based dense maps creation. Comparison of compression techniques with panoramic images," in *Proceedings of the International Conference on Informatics in Control, Automation and Robotics (INSTICC '09)*, pp. 238–246, Milan, Italy, 2009.
- [16] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference*, pp. 23.1–23.6, Manchester, UK, 1988.
- [17] 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.
- [18] L. Payá, L. Fernández, A. Gil, and O. Reinoso, "Map building and Monte Carlo localization using global appearance of omnidirectional images," *Sensors*, vol. 10, no. 12, pp. 11468–11497, 2010.
- [19] A. Friedman, "Framing pictures: the role of knowledge in automated encoding and memory for gist," *Journal of Experimental Psychology: General*, vol. 108, no. 3, pp. 316–355, 1979.
- [20] A. Torralba, "Contextual priming for object detection," *International Journal of Computer Vision*, vol. 53, no. 2, pp. 169–191, 2003.
- [21] 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)*, pp. 886–893, Washington, DC, USA, June 2005.



- [22] F. Amorós, L. Payá, O. Reinoso, L. Fernández, and J. Marín, “Visual map building and localization with an appearance-based approach—comparisons of techniques to extract information of panoramic images,” in *Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics*, pp. 423–426, 2010.
- [23] D. Filliat, “A visual bag of words method for interactive qualitative localization and mapping,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 3921–3926, Roma, Italy, April 2007.

## Research Article

# Feature Selection for Intelligent Firefighting Robot Classification of Fire, Smoke, and Thermal Reflections Using Thermal Infrared Images

Jong-Hwan Kim,<sup>1</sup> Seongsik Jo,<sup>1</sup> and Brian Y. Lattimer<sup>2</sup>

<sup>1</sup>Mechanical & Systems Engineering Department, Korea Military Academy, Seoul, Republic of Korea

<sup>2</sup>Mechanical Engineering Department, Virginia Tech, Blacksburg, VA 24060, USA

Correspondence should be addressed to Jong-Hwan Kim; [jonghwan7028@gmail.com](mailto:jonghwan7028@gmail.com)

Received 26 March 2016; Revised 5 July 2016; Accepted 3 August 2016

Academic Editor: Juan A. Corrales

Copyright © 2016 Jong-Hwan Kim 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.

Locating a fire inside of a structure that is not in the direct field of view of the robot has been researched for intelligent firefighting robots. By classifying fire, smoke, and their thermal reflections, firefighting robots can assess local conditions, decide a proper heading, and autonomously navigate toward a fire. Long-wavelength infrared camera images were used to capture the scene due to the camera's ability to image through zero visibility smoke. This paper analyzes motion and statistical texture features acquired from thermal images to discover the suitable features for accurate classification. Bayesian classifier is implemented to probabilistically classify multiple classes, and a multiobjective genetic algorithm optimization is performed to investigate the appropriate combination of the features that have the lowest errors and the highest performance. The distributions of multiple feature combinations that have 6.70% or less error were analyzed and the best solution for the classification of fire and smoke was identified.

## 1. Introduction

Intelligent firefighting humanoid robots are actively being researched to reduce firefighter injuries and deaths as well as increase their effectiveness on performing tasks [1–5]. One task is locating a fire inside of a structure outside the robot field of view (FOV). Fire, smoke, and their thermal reflections can be clues to determine a heading that will ultimately lead the robot to the fire so that it can suppress it. However, research for accurately classifying these clues has been incomplete.

## 2. Previous Features

In conventional fire (and/or smoke) detection systems [6, 7] in Table 1, temperature, ionization, and ultraviolet light were mainly used to indicate the presence of a fire and/or smoke inside the structure, but they can have a long response time in large spaces [8] and do not provide sufficient data for the location of fire and/or smoke. Recently using vision systems,

color [9–12], motion [13, 14], both [8, 15–17], and texture features [12, 18, 19] have been researched to characterize fire or smoke in Table 1. However, color features from RGB camera are not applicable to firefighting robots due to the fact that RGB cameras may operate in the visible to short wavelength infrared (IR) (less than 1 micron) and are not usable in smoke-filled environments where the visibility has sufficiently decreased [2, 14]. Motion (e.g., dynamical motion, shape changing, etc.) of the feature can be another clue to detect fire and smoke by characterizing flickering flames and smoke flow from a stationary vision system. However, the vision system onboard a robot is moving due to the dynamics of the robot itself, and this causes a large amount of noise that results in extensive computation for motion compensation. Texture features researched in [12, 18, 19] were used to identify fire or smoke. The spatial characteristics of textures can be useful to recognize patterns of fire and smoke by remote sensing and are less influenced by rotation/motion [18].

Long-wavelength infrared cameras, similar to the hand-held thermal infrared cameras (TICs) that are typically used

TABLE 1: Conventional and vision-based features.

Type	Feature	Advantages	Disadvantages
Conventional features [6, 7]	Temperature Ionization UV light	(i) Detect presence of fire and smoke [8]	(i) Long response time [8] (ii) Unable to provide sufficient data for fire locating
Model-based features	Fourier transform [20] Wavelet transform [9]	(i) Frequency content analysis (ii) Flexible analysis of both space and frequency [25]	(i) Unable to be spatially localized [25]
Vision-based features	Color (RGB) [9–12, 26]	(i) Fire (red) (ii) Smoke (gray)	(i) RGB camera cannot function in smoke-filled environments [2, 14]
	Dynamics [13, 14] (motion, shape change, etc.)	(i) Flickering flames recognition (ii) Smoke flow detection	(i) Can be influenced by dynamical robot motion (ii) Expensive computation for motion compensation
	Texture [12, 18, 19, 27]	(i) Spatial characteristics for pattern recognition (ii) Less influenced by rotation and motion [18]	(i) The higher the order texture features, the more the computation
	Feature maps [28] (CNN deep learning)	(i) Superior performance in pattern recognition [29] (ii) Once trained, applicable in real-time	(i) Slow learning speed (ii) GPUs required due to expensive computation

to aid in firefighting tasks within smoke-filled environments [20–22] as well as fire-front and burned-area recognition in remote sensing [23], are used in this research. Due to the fact that TICs absorb infrared radiation in the long-wavelength IR (7–14 microns), they are able to image surfaces even in dense smoke and zero visibility environments [2, 14]. In addition, TIC can provide proper information under local or global darkness, for example, shadows or darkness caused by damaged lighting. Recently, thermal images from TIC are studied to recognize pattern and motion remotely [24]. The cameras will detect hot objects as well as thermal reflections off of surfaces. As a result, image processing on detected objects must be sufficiently robust to discern between desired objects and their thermal reflections.

This study ultimately leads the shipboard autonomous firefighting robot (SAFFiR), whose prototype is displayed in Figure 1, to autonomously navigate toward fire outside FOV in indoor fire environments. For this, the robot needs to identify clues such as smoke and smoke and fire-reflections by itself to correctly navigate toward the fire. However, the recognition of key features has not been fully studied. This paper analyzes appropriate combination of features to accurately classify fire, smoke, their thermal reflections, and other hot objects using thermal infrared images. Large-scale fire tests were conducted to create actual fire environments having various ranges of both temperature and smoke conditions. A long-wavelength IR camera was installed to produce 14-bit thermal images of the fire environment. These images were used to extract motion and statistical texture features in regions of interest (ROI). Bayesian classification was performed to probabilistically identify multiple classes in real-time. To identify the best combination of features for

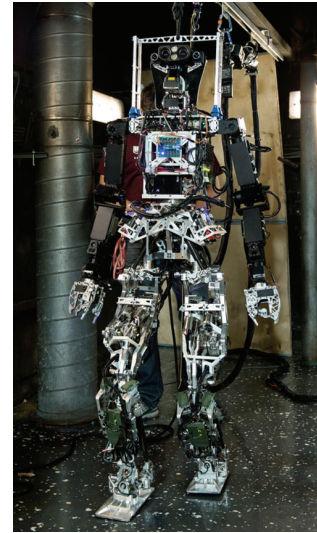


FIGURE 1: A prototype of the shipboard autonomous firefighting robot (SAFFiR). Note that the data used in this paper were not acquired from this platform.

accurate classification, the multiobjective optimization was implemented using two objective functions: resubstitution and cross-validation errors.

### 3. Motion and Texture Features

In pattern recognition system, the choice of features plays an important role in the performance of classification. Both

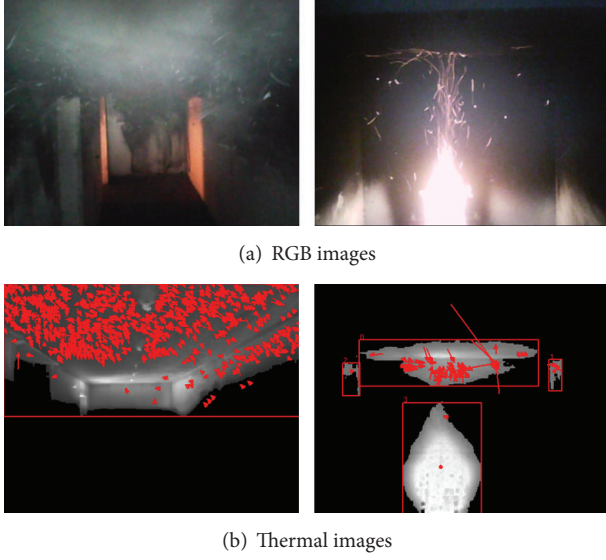


FIGURE 2: (a) RGB images of fire scenes and (b) extracted objects from thermal images with optical flow vectors overlaid.

motion and texture features were selected because they were crucial in the previous study of fire and/or smoke detection and also best suitable for the thermal image analysis that is major information the firefighting robot can acquire under fire environments. Optical flow, a popular motion measurement, was used for the motion features, while the first- and second-statistical texture features were applied for the texture measurement.

A FLIR A35 long-wavelength IR camera, which is capable of imaging through zero visibility environments, was used to produce images. All images were from a  $320 \times 256$ -pixel focal plane array, 60 Hz frame rate that produces 14-bit images with an intensity range of  $-16384$  for  $-40^\circ\text{C}$  to  $-1$  for  $550^\circ\text{C}$ . Fifteen features from optical flow and the statistical texture features are evaluated to find the best feature combination. Optical flow shows temporal variations due to moving objects in the FOV or motion of the robot. The first- and second-order statistical texture features display spatial characteristics of objects in the scene.

**3.1. Motion Features by Optical Flow.** Optical flow is a useful tool to recognize motion of an object in sequential images [30]. It consists of local and global methods. Lucas-Kanade (LK) is a local method that is relatively robust with a less dense flow field, while Horn-Schunck (HS) is a global method with a dense flow field and high sensitivity to noise [31]. Because the intensities in the thermal image change due to the varying fire environment, LK method that has higher robustness compared with HS was selected in this research to measure motion features of the objects. Two features of optical flow vector number (OFVN) and optical flow mean magnitude (OFVMM) were computed to quantitatively characterize motions of fire, smoke, and their reflections. Figure 2 contains RGB and thermal images of dense smoke in a hallway and a wood crib fire in a room. Red arrows in the

thermal images indicate the direction and magnitude of the optical flow vectors with red boxes that show smoke, fire, and thermal reflections.

**3.2. First- and Second-Order Statistical Texture Features.** The first- and second-order statistical features were considered in this study for object classification. The first-order statistical features estimate individual property of pixels, not characterizing any relationship between neighboring pixels, and can be computed using the intensity histogram of the candidate region of interest (ROI) in the image. As described in [32], mean (MNI), variance (VAR), standard deviation (STD), skewness (SKE), and kurtosis (KUR) were calculated by

$$\begin{aligned} \text{MNI} &= \frac{1}{N_P} \sum_{i,j=1}^{N_P} I_{i,j}, \\ \text{VAR} &= \frac{1}{N_P} \sum_{i,j=1}^{N_P} (I_{i,j} - \mu)^2, \\ \text{STD}(\sigma) &= \left( \frac{1}{N_P} \sum_{i,j=1}^{N_P} (I_{i,j} - \mu)^2 \right)^{1/2}, \\ \text{SKE} &= \frac{1}{\sigma^3 N_P} \sum_{i,j=1}^{N_P} (I_{i,j} - \mu)^3, \\ \text{KUR} &= \frac{1}{\sigma^4 N_P} \sum_{i,j=1}^{N_P} (I_{i,j} - \mu)^4, \end{aligned} \quad (1)$$

where  $I_{i,j}$  refers to the intensity of a pixel at  $i$  and  $j$  and  $N_P$  denotes the number of pixels (NOP) of the object in the image. The second-order statistical features represent spatial relationships between a pixel and its neighbors. Gray-level cooccurrence matrix (GLCM) [33] is used to account for adjacent pixel relationships in four directions (horizontal, vertical, left, and right diagonals) by quantizing the spatial cooccurrence of neighboring pixels. A total of seven second-order statistics features were used including dissimilarity (DIS), entropy (ENT), contrast (CON), inverse difference (INV), correlation (COR), uniformity (UNI), and inverse difference moment (IDM). To measure these features, a normalized cooccurrence matrix  $C_{ij}$  is used which can be defined as

$$C_{ij} = \frac{P_{ij}}{\sum_{i,j=1}^{N_G} P_{ij}}, \quad (2)$$

where  $P_{ij}$  refers to the frequency of occurrences of the gray-level of adjacent pixels at  $i$  and  $j$  within the four directions and  $N_G$  denotes the number of the gray-levels in the quantized image. The denominator of (2) normalizes  $P_{ij}$  to be estimates of the cooccurrence probabilities. After building the normalized cooccurrence matrix  $C_{ij}$ , seven features of the second-order statistics features were computed by

$$\begin{aligned} \text{DIS} &= \sum C_{ij} |i - j|, \\ \text{ENT} &= - \sum C_{ij} \log C_{ij}, \end{aligned}$$

$$\begin{aligned}
\text{CON} &= \sum C_{ij} (i - j)^2, \\
\text{INV} &= \sum \frac{C_{ij}}{1 + |i - j|}, \\
\text{COR} &= \sum \frac{(ij + \mu_i \mu_j - \mu_i - \mu_j) C_{ij}}{\sigma_i \sigma_j}, \\
\text{UNI} &= \sum C_{ij}^2, \\
\text{IDM} &= \sum \frac{C_{ij}}{1 + (i - j)^2}.
\end{aligned} \tag{3}$$

#### 4. Object Extraction and Bayesian Classification

One of the main characteristics of fire, smoke, and their thermal reflections in thermal images is that they are higher in intensity than the background. With intensity related to temperature in the thermal image, higher temperature objects appear brighter than the background. Hence, intensity is a primary factor for object extraction from the background. Assuming that the thermal image histogram has a bimodal distribution for foreground (i.e., object) and background, the clustering-based image autothresholding method [34], called Otsu method, can calculate an optimum threshold that separates objects and background creating a binary image with 0 being the background and 1 being the objects. The binary images were filtered to remove small regions and holes inside objects through morphological filtering techniques. After convoluting the original 14-bit image with the filtered-binary image, a final image was obtained that includes the original 14-bit intensities in objects as well as zeroes in the background.

There are several classification methods commonly used in supervised machine learning;  $k$ -nearest neighbors ( $k\text{NN}$ ), decision tree (DT), neural networks (NN), support vector machine (SVM), and Naïve Bayesian. For this study, these classification methods were analyzed by considering three points: capability to classify multiple classes such as fire, smoke, and their thermal reflections; less chance of overfitting problem because, under fire environments, there could be a number of situations that are not learned or trained; real-time implementation because firefighting robot needs to make a decision in real-time; otherwise it cannot operate its task.  $k\text{NN}$  is insensitive to outliers but it needs a large amount of memory and expensive computation [35]. DT has low computation burden but, for the multiclass classification, it may generate a complicated tree structure and may cause overfitting problem [35, 36]. NN shows high performance when processing with multidimensions and continuous features but cannot overcome overfitting problem. SVM provides fast computation and the highest accuracy but it cannot be used for the multilabel classification because it produces binary results [37]. Naïve Bayesian classification is Bayes' theorem-based probabilistic classification and is popular for

pattern recognition applications. Although this method has lower accuracy compared with other classifiers and assumes that each feature is independent, it has fast computation, robustness to untrained cases, and less chance of overfitting [35]. In addition, this classification has the capability of probabilistic decision making over multiple classes with fast computation for real-time implementation. In this study, Bayesian classification is used for evaluation of each feature.

With several given features  $F_1, F_2, \dots, F_q$ , (motion and texture features) we can calculate the probability that one class  $C_h$  (fire, smoke, thermal reflections, etc.) corresponds to the candidate  $k$  by using a conditional probability,  ${}^k p(C_h | F_1 F_2 \dots F_q)$ , also known as the posterior probability. By using Bayes' theorem, it can be written with prior, likelihood, and evidence as shown in

$$\begin{aligned}
{}^k p(C_h | F_1 F_2 \dots F_q) \\
= \frac{{}^k p(C_h) {}^k f(F_1 F_2 \dots F_q | C_h)}{\sum_{C_h} {}^k p(C_h) {}^k f(F_1 F_2 \dots F_q | C_h)},
\end{aligned} \tag{4}$$

where  ${}^k p(C_h)$  is the prior probability, meaning it represents candidate  $k$  probability to be  $C_h$  and can be calculated by number of samples of class  $C_h$  divided by the total number of samples.  ${}^k f(F_1 F_2 \dots F_q | C_h)$  is the likelihood function and the denominator of (4) is the evidence that plays as a normalizing constant by the summation of production between the prior and likelihood at each class. By applying the conditional independence assumption, the likelihood function can be rewritten by

$$f(F_1 F_2 \dots F_q | C_h) = \prod_{i=1}^q f(F_i | C_h). \tag{5}$$

The conditional probability density function  $f(F_i | C_h) \sim N(\mu_{F_i|C_h}, \Sigma_{F_i|C_h})$  can be described as

$$f(F_i | C_h) = \frac{1}{\sqrt{2\pi\Sigma_{F_i|C_h}}} e^{-(1/2)(F_i - \mu_{F_i|C_h})^T \Sigma_{F_i|C_h}^{-1} (F_i - \mu_{F_i|C_h})}, \tag{6}$$

where

$$\begin{aligned}
\mu_{F_i|C_h} &= \mu_{F_i} + \Sigma_{C_h F_i}^T \Sigma_{C_h C_h}^{-1} (C_h - \mu_{C_h}), \\
\Sigma_{F_i|C_h} &= \Sigma_{F_i F_i} - \Sigma_{C_h F_i}^T \Sigma_{C_h C_h}^{-1} \Sigma_{C_h F_i}.
\end{aligned} \tag{7}$$

As shown in Table 2, Gaussian parameters for fifteen features with respect to smoke, smoke thermal reflection, fire, and fire thermal reflection were estimated by using the maximum likelihood estimation [38]. Probability density distributions for the entire features are illustrated in Figure 3. With (5), the evidence and then the posterior probability of each class were calculated. By applying the maximum priority decision rule in (8), the Bayesian classification was used to predict the class and probability of each candidate in the scene:

$$\begin{aligned}
\text{class} &= \underset{C_h}{\text{argmax}} ({}^k p(C_h | F_1 F_2 \dots F_q)), \\
\text{prob} &= \max ({}^k p(C_h | F_1 F_2 \dots F_q)).
\end{aligned} \tag{8}$$



TABLE 2: Gaussian parameters.

	Smoke		Smoke-reflection		Fire		Fire-reflection	
	$\mu$	$\Sigma$	$\mu$	$\Sigma$	$\mu$	$\Sigma$	$\mu$	$\Sigma$
MNI	$-1.2665E+04$	$6.9008E+02$	$-1.3383E+04$	$3.5543E+02$	$-5.9714E+03$	$1.6050E+03$	$-7.1399E+03$	$4.6070E+02$
VAR	$4.7578E+05$	$4.4154E+05$	$4.4012E+04$	$4.1227E+04$	$1.0501E+07$	$2.2343E+06$	$1.3300E+06$	$8.8178E+05$
NOP	$1.0733E+03$	$4.2287E+03$	$3.2220E+01$	$1.3676E+02$	$2.2174E+02$	$1.7105E+03$	$5.9575E+01$	$2.3911E+02$
STD	$6.2146E+02$	$2.9931E+02$	$1.8314E+02$	$1.0236E+02$	$3.2170E+03$	$3.8930E+02$	$1.0534E+03$	$4.6992E+02$
SKE	$1.1672E-01$	$5.8208E-01$	$-9.4009E-02$	$6.2857E-01$	$2.2230E-02$	$5.2287E-01$	$2.2385E-01$	$7.1380E-01$
KUR	$3.0045E+00$	$2.0213E+00$	$3.6553E+00$	$1.6131E+00$	$2.2283E+00$	$1.1517E+00$	$3.4848E+00$	$1.1912E+00$
OFVN	$2.8832E+04$	$1.4156E+04$	$1.1848E+03$	$8.7345E+02$	$1.1257E+04$	$1.3722E+04$	$2.9978E+03$	$2.2034E+03$
OFVMM	$8.6830E+01$	$5.6259E+01$	$1.1687E+02$	$7.9444E+01$	$1.7598E+02$	$2.9308E+02$	$1.2641E+02$	$6.7234E+01$
DIS	$7.1791E-02$	$1.7966E-02$	$2.8045E-02$	$1.2245E-02$	$1.0937E-01$	$8.6515E-02$	$4.7308E-02$	$2.8550E-02$
ENT	$4.1949E-01$	$9.3683E-02$	$1.9576E-01$	$8.6106E-02$	$3.3681E-01$	$1.9107E-01$	$1.7115E-01$	$9.9702E-02$
CON	$8.9384E-01$	$3.6871E-01$	$1.2402E-01$	$6.5921E-02$	$9.2956E-01$	$6.9179E-01$	$3.9683E-01$	$2.6996E-01$
IND	$9.8588E-01$	$6.3845E-03$	$9.9646E-01$	$1.5141E-03$	$9.6334E-01$	$3.7231E-02$	$9.8771E-01$	$7.9275E-03$
COR	$9.6636E-01$	$3.1625E-02$	$8.8533E-01$	$4.3467E-02$	$8.9406E-01$	$3.0935E-02$	$8.8417E-01$	$5.7861E-02$
UNI	$5.1044E-01$	$2.0109E-01$	$9.5222E-01$	$3.1669E-02$	$6.7906E-01$	$2.6718E-01$	$8.5305E-01$	$1.0599E-01$
IDM	$6.0061E-01$	$1.9160E-01$	$9.7533E-01$	$1.6844E-02$	$7.7850E-01$	$2.4321E-01$	$9.2043E-01$	$5.8631E-02$

TABLE 3: The object numbers of smoke, smoke-reflection, fire, fire-reflection, and other hot objects classes.

Type	Total	Smoke	Smoke-reflection	Fire	Fire-reflection	Other hot objects
Number of objects	10,775	5190	1445	1464	489	2187

Figure 3 shows probability density distribution of each class using the Gaussian parameters of Table 2. Gaussian distribution for classes in Figure 3 shows how fire, fire-reflection, smoke, and smoke-reflection are distributed by the fifteen features. Some features split out the distribution of the four classes while others cause overlap. For example, MNI best describes a well split out case of the classes, although smoke and its reflection and fire and its reflection do overlap. SKE shows the worst case in which all classes overlap making it impossible to distinguish any of the four classes.

## 5. Result and Discussion

The accuracy in classifying fire objects was analyzed using data from a series of large-scale tests in the facility [1] using actual fires up to 75 kW. Fires included latex foam, wood cribs, and propane gas fires from a sand burner. These different types of fires produced a range of temperature and smoke conditions. Latex foam fires produced lower temperature conditions but dense, low visibility smoke. Conversely, propane gas fires produced higher gas temperatures and light smoke. Wood crib fires resulted in smoke and gas temperatures between those of latex foam and propane gas fires; however, these fires resulted in sparks created from the burning wood. Thermal images were collected by driving a wheeled mobile robot through the setup during a fire test. A total of 10,775 objects were collected from the experiments and categorized as either smoke, smoke-reflection, fire, fire-reflection, or other hot objects in order to be served as clues to lead the firefighting robot to navigate toward the fire source

outside the FOV. In addition, as each object has sixteen corresponding data points (fifteen features and a class), the total number of data points used in this paper is 172,400. The numbers of each object in this experiment are shown in Table 3.

Two types of error criteria (resubstitution and  $k$ -fold cross-validation errors [39]) were used to measure how each feature accurately performs in the classification. Resubstitution error takes the entire dataset to compare the actual classes with the predicted classes by the Bayesian classification in order to examine how well the actual and predicted classes match each other. When this criterion is used alone to enhance accuracy, the classification can be overfitted to the training dataset. Cross-validation error is advantageous to detect and prevent from overfitting. Instead of using the entire dataset, cross-validation randomly selects and splits the dataset into  $k$  partitions of approximately equal size ( $k = 10$ ) to estimate a mean error by comparing between the randomly selected partition and trained results of the remaining partitions.

**5.1. Single Feature Performance.** The performance results of each feature are shown in Table 4. The first-order statistical texture features MNI, VAR, and STD produced the lowest errors while NOP, SKE, and KUR show the highest. These results show that MNI and VAR are beneficial to distinguish fire, smoke, and thermal reflections while motion features are not. As NOP shows the highest error, OFVMM, one of the motion features, shows the second highest errors compared with the other features. This is in part attributed to the dynamic motion of the robot. ENT and COR second-order

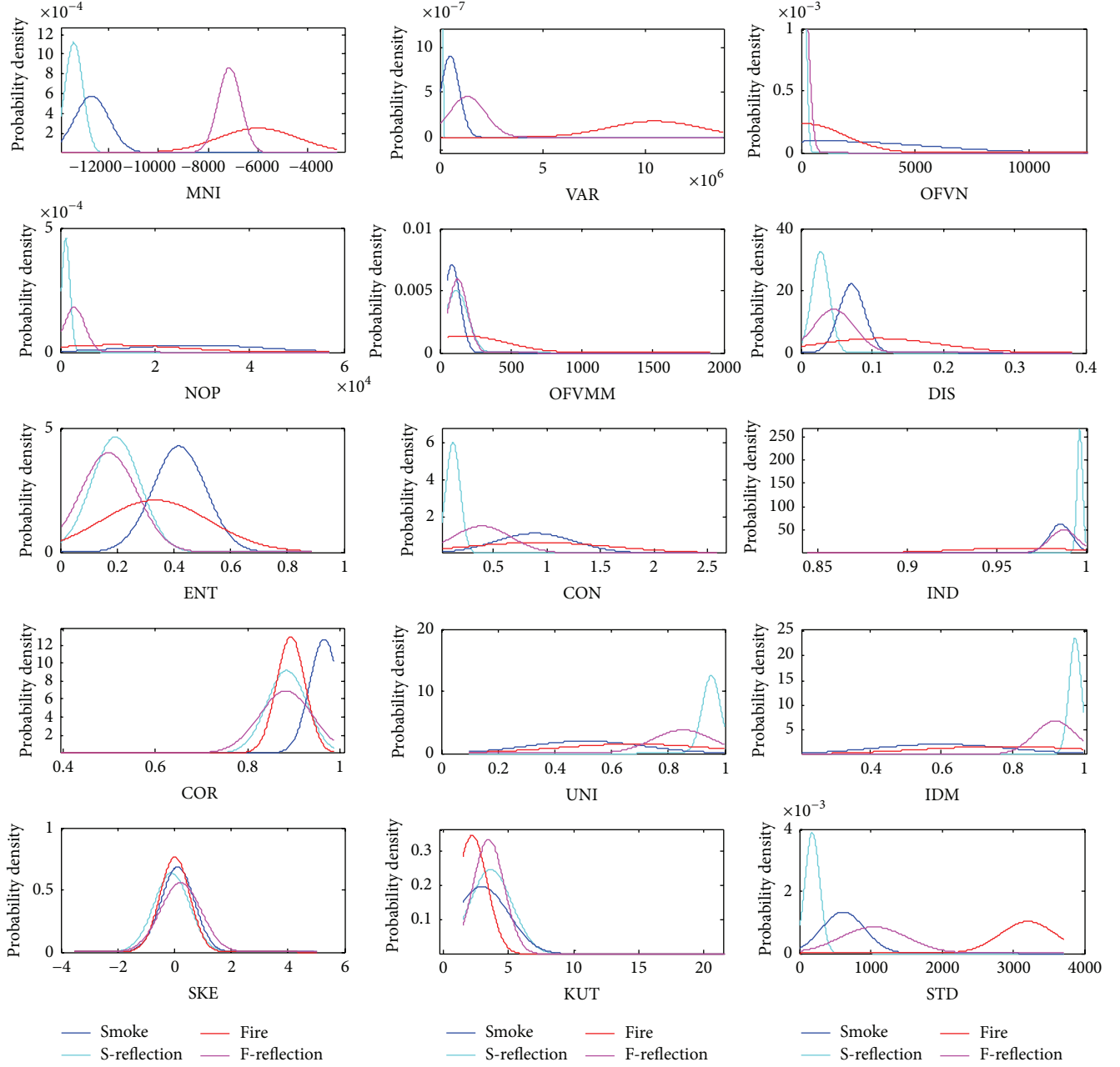


FIGURE 3: Probability density distributions of each feature.

statistical texture features show 42~45% error, which is higher than the other second-order features.

**5.2. Multiple Feature Combination Performance.** The error results in Table 4 demonstrate that a single feature cannot accurately classify fire, smoke, and thermal reflections. Thus, possible combination of multiple features was considered and analyzed to find the best combination of the features. The total number of all possible combinations that have two or more features is

$$N_{\text{total}} = \sum_{z=2}^m \binom{m}{z}, \quad (9)$$

where  $m$  refers to the total number of features (i.e.,  $m = 15$ ) and  $z$  is the number of features in the combination. Based on all possible combination, the multiobjective genetic algorithm optimization [40] in the global optimization toolbox of MATLAB was used to find the best combination of features that has the highest performance in the classification. The objective functions in the optimization, resubstitution and  $k$ -fold cross-validation errors [39], were used to measure how accurately different feature combinations perform in the classification.

Figure 4 contains a plot of the error associated with the most promising feature combinations. The behavioral solution set is defined as feature combinations with less than 7%

TABLE 4: Performance of each feature.

	Resubstitution error (%)	Cross-validation error (%)
MNI	23.7	23.8
VAR	24.3	24.3
NOP	72.1	72.0
STD	23.2	23.2
SKE	52.7	52.8
KUR	50.6	50.6
OFVN	39.5	40.0
OFVMM	58.6	58.6
DIS	29.0	29.0
ENT	44.6	44.6
CON	28.5	28.5
IND	30.1	30.1
COR	41.2	41.1
UNI	34.5	34.5
IDM	37.7	37.7

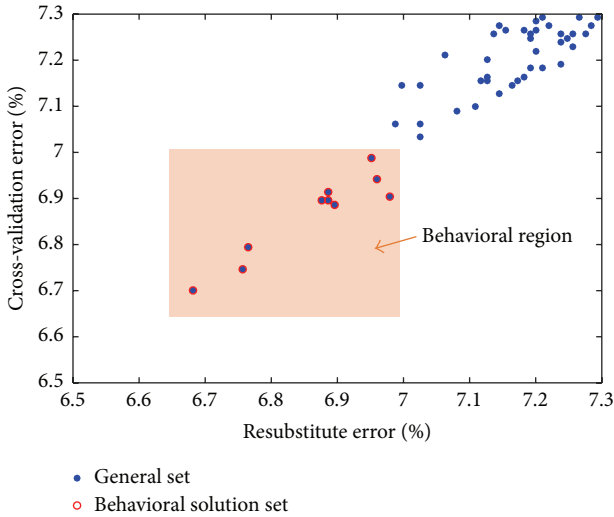


FIGURE 4: Multiobjective optimization result showing the general set and behavioral solution set (colored region).

error for both objective functions while the general set refers to all other possible feature combinations. The behavioral solution set contains 0.0061% of all possible feature combinations.

The occurrence probability of features in the behavioral solution set is illustrated in Figure 5. In the behavioral solution set, the first-order statistic texture features MNI and SKE always exist while OFVN, NOP, and OFVMM features do not. Both the first-order statistical texture features STD and VAR and the second-order statistical texture features COR, ENT, and DIS show a higher occurrence compared with the other first- and second-order texture features while KUR, IDM, UNI, IND, and CON show lower occurrence. Note that, due to the robot's dynamical motion, motion features were not successful and even not included in the top 10 feature combinations of the behavioral set.

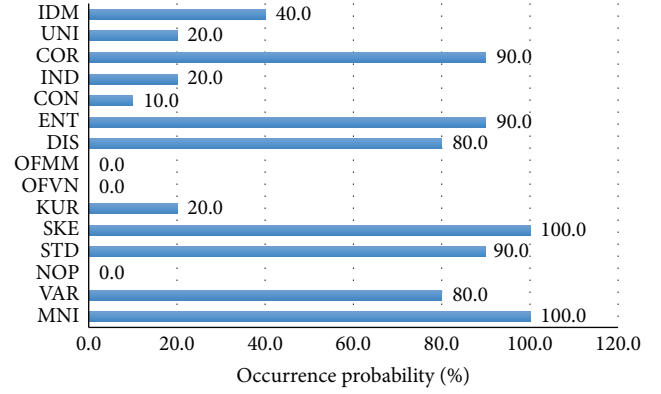


FIGURE 5: Occurrence analysis of the features in the behavioral region.

The top features based on the probability occurrence in Figure 5 are COR, ENT, DIS, SKE, STD, VAR, and MNI. However, the combination of these seven features does not result in the best solution for classification. Table 5 contains the classification performance of the combination of features in the behavioral solution set. In order to evaluate the performance of each feature combination, various performance measures have been used such as precision, sensitivity,  $F$ -measure, and accuracy. Precision measures the fraction of positive instances from the group that the classifier predicted to be positive, and recall measures the fraction of positive examples from the positive group of the actual class and [35].  $F$ -measure is the harmonic mean, and accuracy is the proportion of true results. These measures can be mathematically defined as

$$\text{Precision} = \frac{TP}{TP + FP},$$

$$\text{Sensitivity} = \frac{TP}{TP + FN},$$

$$F\text{-Measure} = \frac{2(\text{Sensitivity} \cdot \text{Precision})}{\text{Sensitivity} + \text{Precision}}, \quad (10)$$

$$\text{Accuracy} = \frac{TP}{TP + FP + FN},$$

where TP is correctly classified positive cases, FP is incorrectly classified negative cases, and FN is incorrectly classified positive cases. For the performance measurement, confusion matrixes were created as described in Appendix and applied into (10). In the precision, index number 1 combination shows the highest performance in the behavioral solution set while index number 7 combination shows the lowest. In the sensitivity, index number 7 combination records the highest results while index number 4 does the lowest. In the  $F$ -measure and accuracy, index number 2 combination shows the highest record while index number 4 does the lowest. Based on the confusion matrixes, most of misclassification occurs in the classification of smoke, smoke-reflection, and other hot objects, because, during small fire, texture patterns of these classes were diminished and the intensity was too low

TABLE 5: Results of error, case, and performance at each feature combination in the behavioral solution set (Resu. means resubstitution and Cros. refers to cross-validation).

Index	Combination of features	Error (%)		Case			Performance (%)			
		Resu.	Cros.	TP	FP	FN	Precision	Sensitivity	<i>F</i> -measure	Accuracy
1	MNI, VAR, ENT, COR, SKE	6.90	6.89	10049	402	324	<b>96.15</b>	96.88	96.51	93.26
2	MNI, DIS, COR, SKE, STD	6.68	6.70	10069	459	247	95.64	97.61	<b>96.62</b>	<b>93.45</b>
3	MNI, ENT, COR, SKE, STD	6.76	6.75	10061	447	267	95.75	97.41	96.57	93.37
4	MNI, VAR, DIS, ENT, CON, COR, SKE, STD	6.98	6.90	9980	454	341	95.65	96.70	96.17	92.62
5	MNI, VAR, DIS, ENT, COR, UNI, SKE, STD	6.89	6.90	10037	453	285	95.68	97.24	96.45	93.15
6	MNI, VAR, DIS, ENT, COR, IDM, SKE, STD	6.77	6.79	10047	461	267	95.61	97.41	96.50	93.24
7	MNI, VAR, DIS, ENT, IDM, SKE, KUR, STD	6.96	6.94	10033	505	237	95.21	<b>97.69</b>	96.43	93.11
8	MNI, VAR, DIS, ENT, IND, COR, UNI, SKE, STD	6.95	6.99	10029	451	295	95.70	97.14	96.41	93.08
9	MNI, VAR, DIS, ENT, IND, COR, IDM, SKE, STD	6.88	6.90	10035	457	283	95.64	97.26	96.44	93.13
10	MNI, VAR, DIS, ENT, COR, IDM, SKE, KUR, STD	6.89	6.91	10036	478	261	95.45	97.47	96.45	93.14

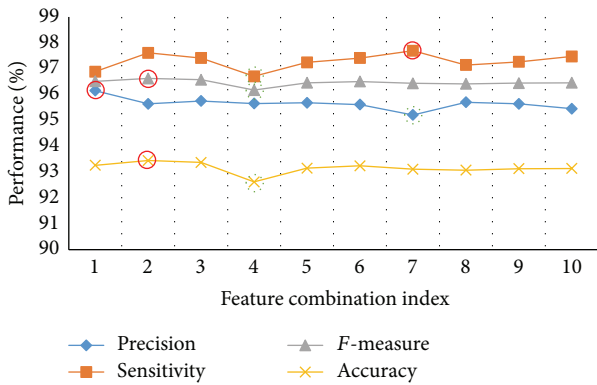


FIGURE 6: Results of performance at each feature combination in the behavioral solution set.

to distinguish. The best solution was determined to be index number 2 combination of MNI, DIS, COR, SKE, and STD, which has the lowest of resubstitution and cross-validation errors, 6.68% and 6.70%, respectively. This combination includes all of the top features based on the probability occurrence except ENT and VAR. The four performance results at each feature combination in the behavioral solution set are shown in Figure 6 where the highest results are marked in red circles and the lowest in green-dot circles. Sensitivity appears higher than precision at each feature combination because FPs are larger than FNs in the confusion matrix. Particularly, index number 7 has the biggest difference between FP and FN resulting in the highest sensitivity and lowest precision. The summation of FP and FN in index number 4 is the highest in the behavioral solution set resulting in the lowest accuracy while index number 2 has the lowest summation of FP and FN providing the highest accuracy.

This study investigated a wide range of features from long-wavelength infrared camera images, analyzed normal distributions of fifteen features with respect to the classes of smoke, fire, and their thermal reflections, and discovered the highest performing feature combination by examining single features and multiple feature combinations. As a result, the proposed feature combination of MNI, DIS, COR, SKE, and STD increases the performance compared with the previous study [1] which used MNI, VAR, ENT, and IDM. As shown in Figure 7, the errors are reduced by 2.86% and 2.68% resubstitution and cross-validation errors and performances are increased by 2.90%, 1.58%, 0.20%, and 2.85%, accuracy, *F*-measure, sensitivity, and precision, respectively.

Figure 8 shows original visual and thermal images with the robot at three different locations: start point, hallway entrance, and room entrance described in the experimental facility. Each row relates to a series of images from the robot at three locations. The first row contains visible images of the robot view. As seen in the visible image at start point, further information regarding the hallway is limited due to shadowing of the light. The image at hallway entrance shows a smoke layer in the upper portion of the hallway due to a fire inside the room. The image at the room entrance displays a wood crib fire with sparks. Because of soot and relative difference in brightness, the background is shown darker and thus limiting information on the background around the fire.

Thermal infrared images are displayed in the second row to show information that RGB camera cannot provide in fire environments. Unlike visual image at start point that is obscured due to shadowing, the presence of smoke and its thermal reflections on the ventilation hood can be obviously perceived. The red boxes on thermal images indicate objects extracted through the adaptive object extraction with optical flows and identification numbers. In spite of dense

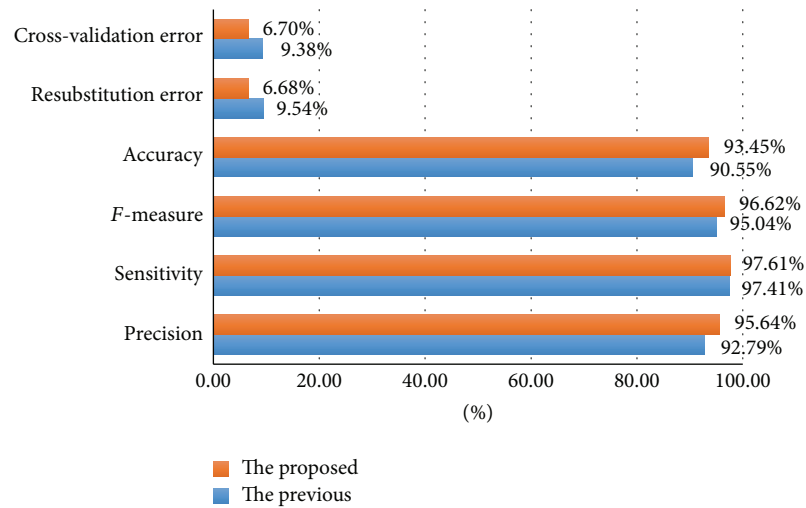


FIGURE 7: Result comparison between the previous and the proposed studies.

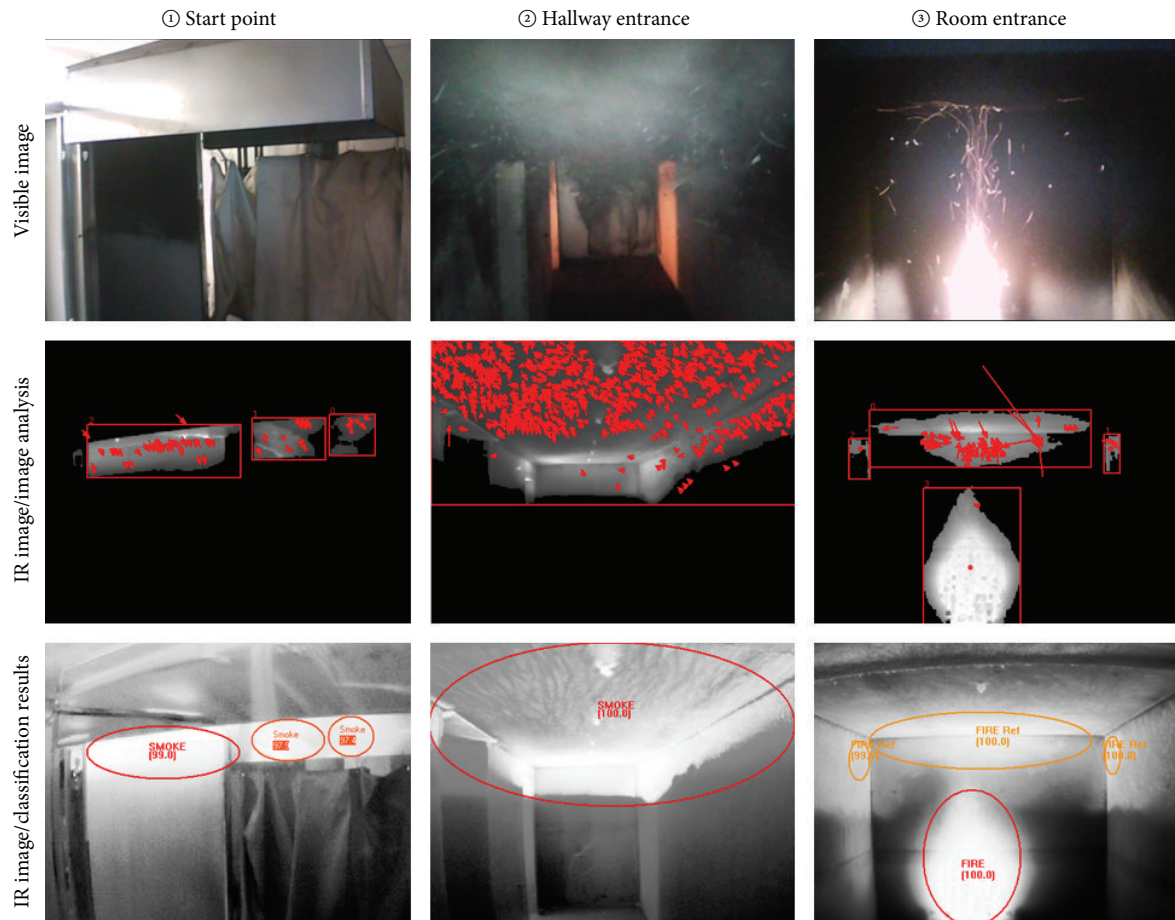


FIGURE 8: Original visible and IR images with image analysis and classification results at different locations in the test setup of actual fires.

smoke-filled and low visibility environments, thermal images can generate the images of smoke and fire, as well as background information that is otherwise not visible through visual imaging.

On the third row, class labels and posterior probabilities of each candidate are displayed at the center of candidate ROI as a result of Bayesian classification. Using enhanced image processing techniques, the thermal images can be more



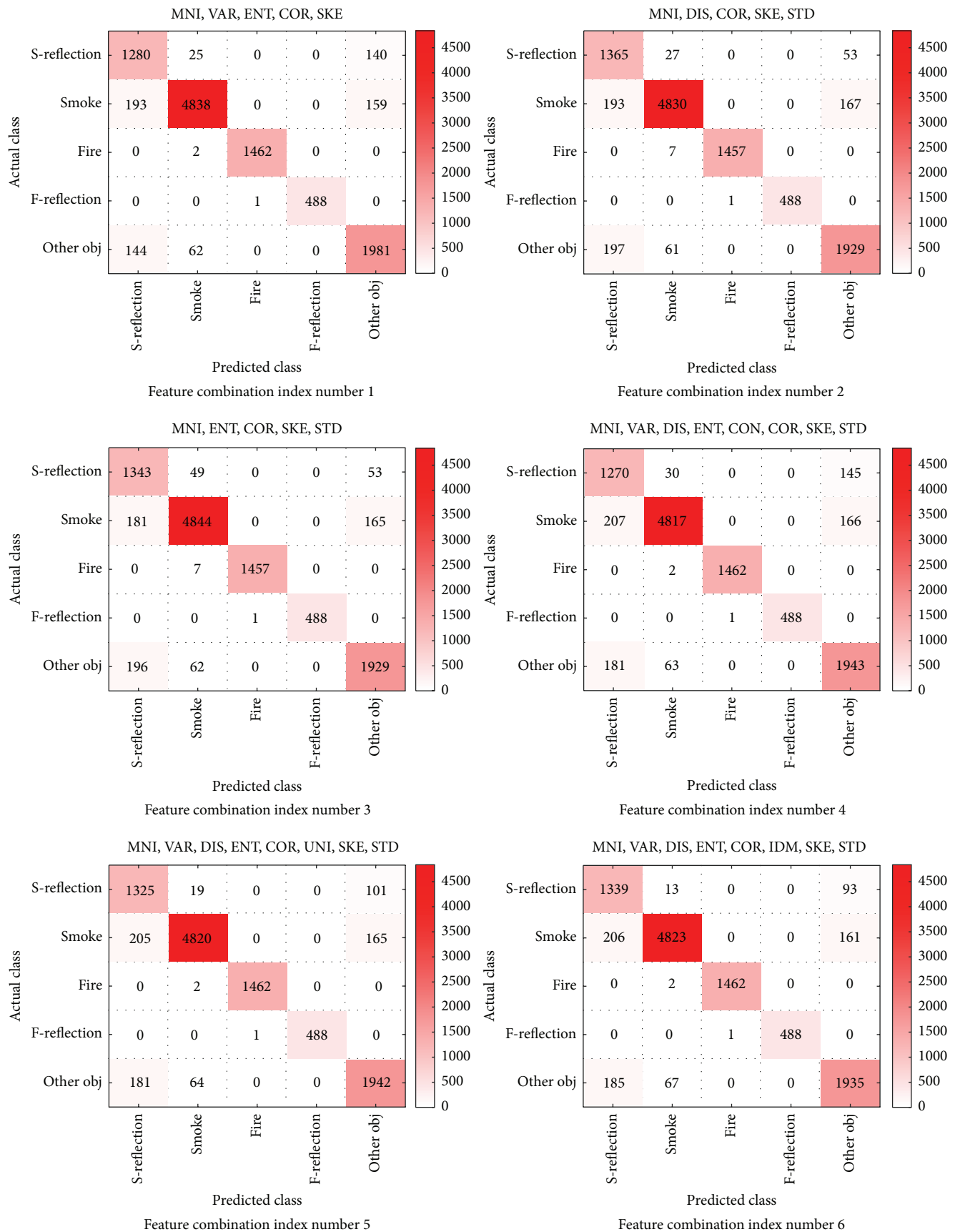


FIGURE 9: Continued.



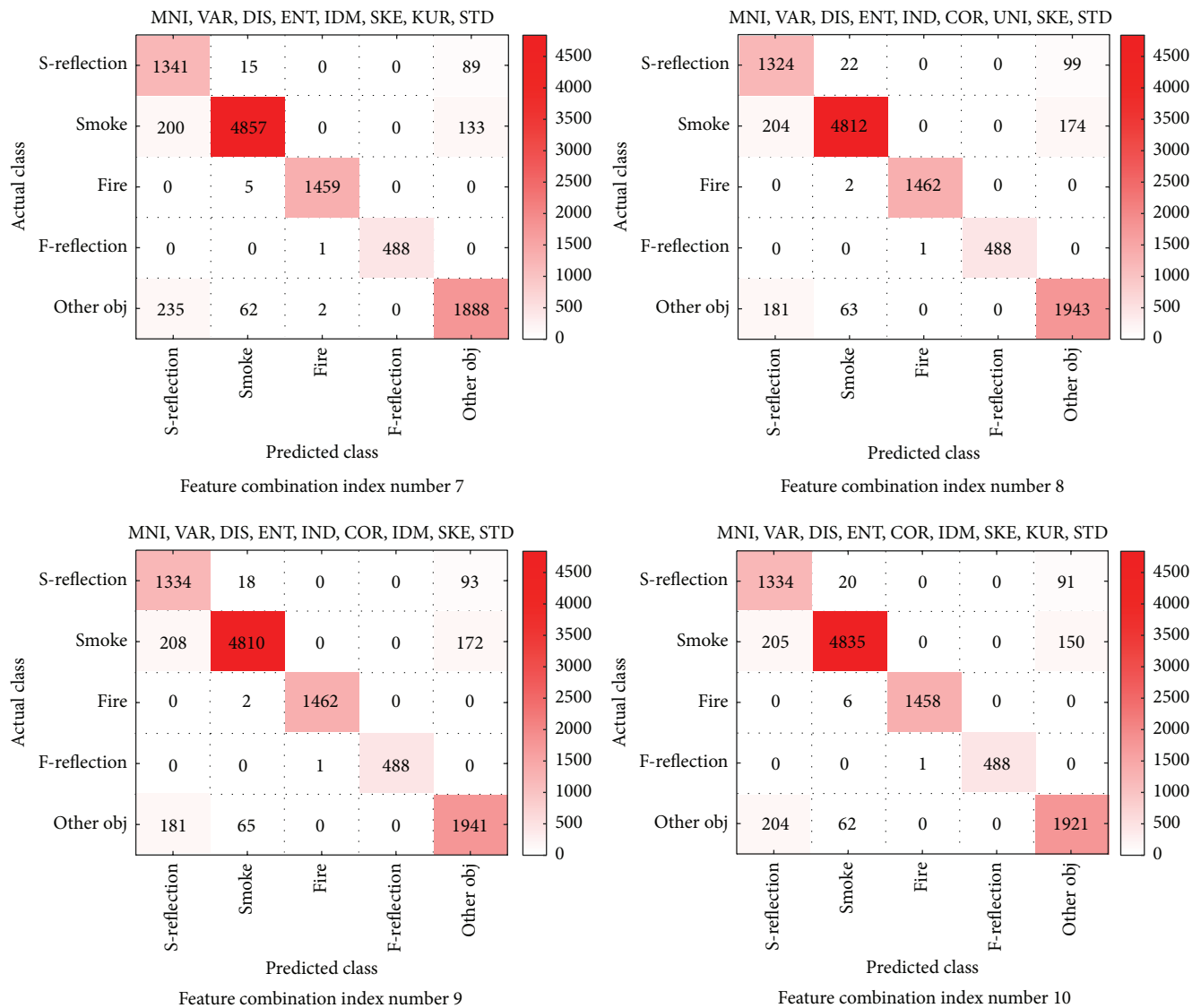


FIGURE 9: Results of confusion matrix at each feature combination.

refined and clearer than the thermal images on the second row. Smoke, fire, and their thermal reflections are identified and marked in red or orange ellipses.

## 6. Conclusion

The appropriate combination of features was investigated to accurately classify fire, smoke, and their thermal reflections using thermal images. Gray-scale 14-bit images from a single infrared camera were used to extract motion and texture features by applying a clustering-based, autothresholding technique. Bayesian classification is performed to probabilistically identify multiple classes during real-time implementation. To find the best combination of features, a multi-objective genetic algorithm optimization was implemented using resubstitution and cross-validation errors as objective functions. Large-scale fire tests with different fire sources

were conducted to create a range of temperature and smoke conditions to evaluate the feature combinations.

Fifteen motion and texture features were analyzed and the probability density functions of the features were computed by the maximum likelihood estimation. The combination of multiple features was determined to more accurately classify fire, smoke, and thermal reflections compared with a single feature. In the behavioral solution set where feature combinations produce less than 7% resubstitution and cross-validation errors, COR, ENT, DIS, SKE, STD, VAR, and MNI had 80.0% or more occurrence while other features had 40.0% or less occurrence. The feature combination of MNI, DIS, COR, SKE, and STD produced the highest performance in the classification resulting in 6.68% and 6.70%, resubstitution and cross-validation errors, and 95.64%, 97.61%, 96.62%, and 93.45%, precision, sensitivity, *F*-measure, and accuracy, respectively.

In the near future, the classification of fire, smoke, and their thermal reflections will be evaluated on any classifiers and features to increase performance. The convolution neural network of deep learning which has recently shown high performance could be explored as a classifier; also model-based image features such as discrete wavelet transform will be further studied.

## Appendix

See Figure 9.

## Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this manuscript.

## Acknowledgments

This work was sponsored by the Office of Naval Research Grant no. N00014-11-1-0074 scientific office Dr. Thomas McKenna in USA, Hwarang-dae Research Institute in Seoul, and Agency for Defense Development in Daejeon, South Korea. The authors would like to thank Mr. Joseph Starr and Mr. Josh McNeil for assisting in performing the fire tests. The authors would also like to thank Rosana K. Lee for helping and supporting this research.

## References

- [1] J.-H. Kim and B. Y. Lattimer, "Real-time probabilistic classification of fire and smoke using thermal imagery for intelligent firefighting robot," *Fire Safety Journal*, vol. 72, pp. 40–49, 2015.
- [2] J. W. Starr and B. Y. Lattimer, "Evaluation of navigation sensors in fire smoke environments," *Fire Technology*, vol. 50, no. 6, pp. 1459–1481, 2014.
- [3] J.-H. Kim, B. Keller, and B. Y. Lattimer, "Sensor fusion based seek-and-find fire algorithm for intelligent firefighting robot," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '13)*, pp. 1482–1486, IEEE, Wollongong, Australia, July 2013.
- [4] J. G. McNeil, J. Starr, and B. Y. Lattimer, "Autonomous fire suppression using multispectral sensors," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics: Mechatronics for Human Wellbeing (AIM '13)*, pp. 1504–1509, Wollongong, Australia, July 2013.
- [5] J.-H. Kim, J. W. Starr, and B. Y. Lattimer, "Firefighting robot stereo infrared vision and radar sensor fusion for imaging through smoke," *Fire Technology*, vol. 51, no. 4, pp. 823–845, 2015.
- [6] R. C. Luo and K. L. Su, "Autonomous fire-detection system using adaptive sensory fusion for intelligent security robot," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 3, pp. 274–281, 2007.
- [7] M. A. Jackson and I. Robins, "Gas sensing for fire detection: measurements of CO, CO<sub>2</sub>, H<sub>2</sub>, O<sub>2</sub>, and smoke density in European standard fire tests," *Fire Safety Journal*, vol. 22, no. 2, pp. 181–205, 1994.
- [8] B. U. Töreyn, R. G. Cinbiş, Y. Dedeoğlu, and A. E. Çetin, "Fire detection in infrared video using wavelet analysis," *Optical Engineering*, vol. 46, no. 6, Article ID 067204, 2007.
- [9] B. U. Töreyn, Y. Dedeoğlu, and A. E. Çetin, "Wavelet based real-time smoke detection in video," in *Proceedings of the 13th European Signal Processing Conference*, pp. 4–8, Antalya, Turkey, September 2005.
- [10] T. Celik, H. Demirel, H. Ozkaramanli, and M. Uyguroglu, "Fire detection using statistical color model in video sequences," *Journal of Visual Communication and Image Representation*, vol. 18, no. 2, pp. 176–185, 2007.
- [11] L. Merino, F. Caballero, J. R. Martínez-de-Dios, I. Maza, and A. Ollero, "An unmanned aircraft system for automatic forest fire monitoring and measurement," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1, pp. 533–548, 2012.
- [12] Y. Wang, T. W. Chua, R. Chang, and N. T. Pham, "Real-time smoke detection using texture and color features," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR '12)*, pp. 1727–1730, Tsukuba, Japan, November 2012.
- [13] G. Marbach, M. Loepfe, and T. Brupbacher, "An image processing technique for fire detection in video images," *Fire Safety Journal*, vol. 41, no. 4, pp. 285–289, 2006.
- [14] M. I. Chacon-Murguia and F. J. Perez-Vargas, "Thermal video analysis for fire detection using shape regularity and intensity saturation features," in *Pattern Recognition*, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, C. B.-Y. Brants, and E. R. Hancock, Eds., vol. 6718 of *Lecture Notes in Computer Science*, pp. 118–126, Springer, Berlin, Germany, 2011.
- [15] W. Phillips III, M. Shah, and N. D. V. Lobo, "Flame recognition in video," *Pattern Recognition Letters*, vol. 23, no. 1–3, pp. 319–327, 2002.
- [16] D. Han and B. Lee, "Development of early tunnel fire detection algorithm using the image processing," in *Advances in Visual Computing*, pp. 39–48, Springer, Berlin, Germany, 2006.
- [17] Y. Chunyu, F. Jun, W. Jinjun, and Z. Yongming, "Video fire smoke detection using motion and color features," *Fire Technology*, vol. 46, no. 3, pp. 651–663, 2010.
- [18] F. Yuan, "Video-based smoke detection with histogram sequence of LBP and LBPV pyramids," *Fire Safety Journal*, vol. 46, no. 3, pp. 132–139, 2011.
- [19] F. Lafarge, X. Descombes, and J. Zerubia, "Textural kernel for SVM classification in remote sensing: application to forest fire detection and Urban area extraction," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '05)*, pp. 1096–1099, September 2005.
- [20] F. Amon and A. Ducharme, "Image frequency analysis for testing of fire service thermal imaging cameras," *Fire Technology*, vol. 45, no. 3, pp. 313–322, 2009.
- [21] F. Amon, V. Benetis, J. Kim, and A. Hamins, "Development of a performance evaluation facility for fire fighting thermal imagers," in *Defense and Security*, pp. 244–252, 2004.
- [22] F. D. Maxwell, "A portable IR system for observing fire through smoke," *Fire Technology*, vol. 7, no. 4, pp. 321–331, 1971.
- [23] A. Barducci, D. Guzzi, P. Marcoionni, and I. Pippi, "Infrared detection of active fires and burnt areas: theory and observations," *Infrared Physics & Technology*, vol. 43, no. 3–5, pp. 119–125, 2002.
- [24] C. Wang and S. Qin, "Adaptive detection method of infrared small target based on target-background separation via robust principal component analysis," *Infrared Physics & Technology*, vol. 69, pp. 123–135, 2015.

- [25] N. Aggarwal and R. K. Agrawal, "First and second order statistics features for classification of magnetic resonance brain images," *Journal of Signal and Information Processing*, vol. 3, no. 2, pp. 146–153, 2012.
- [26] B. Ko, K.-H. Cheong, and J.-Y. Nam, "Early fire detection algorithm based on irregular patterns of flames and hierarchical Bayesian Networks," *Fire Safety Journal*, vol. 45, no. 4, pp. 262–270, 2010.
- [27] H. Maruta, Y. Kato, A. Nakamura, and F. Kurokawa, "Smoke detection in open areas using its texture features and time series properties," in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE '09)*, pp. 1904–1908, IEEE, Seoul, South Korea, July 2009.
- [28] C. M. Bautista, C. A. Dy, M. I. Mañalac, R. A. Orbe, and M. Cordel, "Convolutional neural network for vehicle detection in low resolution traffic videos," in *Proceedings of the IEEE Region 10 Symposium (TENSYP)*, pp. 277–281, IEEE, Bali, Indonesia, May 2016.
- [29] H. Wang, Y. Cai, X. Chen, and L. Chen, "Night-time vehicle sensing in far infrared image with deep learning," *Journal of Sensors*, vol. 2016, Article ID 3403451, 8 pages, 2016.
- [30] C. Shen, Z. Bai, H. Cao et al., "Optical flow sensor/INS/magnetometer integrated navigation system for MAV in GPS-denied environment," *Journal of Sensors*, vol. 2016, Article ID 6105803, 10 pages, 2016.
- [31] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 1–21, 2005.
- [32] A. S. N. Huda and S. Taib, "Suitable features selection for monitoring thermal condition of electrical equipment using infrared thermography," *Infrared Physics and Technology*, vol. 61, pp. 184–191, 2013.
- [33] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.
- [34] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, pp. 23–27, 1975.
- [35] P. Harrington, *Machine Learning in Action*, Manning Publications, 2012.
- [36] D. J. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*, MIT Press, 2001.
- [37] D. Lin, X. Xu, and F. Pu, "Bayesian information criterion based feature filtering for the fusion of multiple features in high-spatial-resolution satellite scene classification," *Journal of Sensors*, vol. 2015, Article ID 142612, 10 pages, 2015.
- [38] F. Van Der Heijden, R. Duin, D. De Ridder, and D. M. Tax, *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB*, John Wiley & Sons, 2005.
- [39] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, Montreal, Canada, August 1995.
- [40] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 16, John Wiley & Sons, New York, NY, USA, 2001.

## Research Article

# Monte Carlo Registration and Its Application with Autonomous Robots

**Christian Rink, Simon Kriegel, Daniel Seth, Maximilian Denninger, Zoltan-Csaba Marton, and Tim Bodenmüller**

*Institute of Robotics and Mechatronics, German Aerospace Center, 82234 Oberpfaffenhofen, Germany*

Correspondence should be addressed to Christian Rink; christian.rink@dlr.de

Received 25 March 2016; Revised 28 June 2016; Accepted 10 July 2016

Academic Editor: Pablo Gil

Copyright © 2016 Christian Rink 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 work focuses on Monte Carlo registration methods and their application with autonomous robots. A streaming and an offline variant are developed, both based on a particle filter. The streaming registration is performed in real-time during data acquisition with a laser striper allowing for on-the-fly pose estimation. Thus, the acquired data can be instantly utilized, for example, for object modeling or robot manipulation, and the laser scan can be aborted after convergence. Curvature features are calculated online and the estimated poses are optimized in the particle weighting step. For sampling the pose particles, uniform, normal, and Bingham distributions are compared. The methods are evaluated with a high-precision laser striper attached to an industrial robot and with a noisy Time-of-Flight camera attached to service robots. The shown applications range from robot assisted teleoperation, over autonomous object modeling, to mobile robot localization.

## 1. Introduction

Pose estimation is needed in a lot of different robotic applications, such as object pose estimation for grasping or manipulating objects, mobile robot localization, or registration of submodels in 3D modeling. In many cases a streaming pose estimation is beneficial or mandatory. One example is autonomous object modeling: if an object is placed on a table or other objects are in the proximity, the bottom or occluded part cannot be modeled without repositioning the object. However, if the object is repositioned and a registration is performed with newly acquired data, the autonomous 3D modeling could continue in order to acquire a complete object model. A streaming pose estimation after object repositioning has various positive impacts on the entire approach. First, time is saved as the pose estimation is readily available after data acquisition. Second, streaming pose estimation has the potential of reporting convergence or failure during data acquisition, enabling an autonomous reaction of the robot, for example, switching to modeling or rescanning. Third, after convergence, acquired data can be passed on-the-fly to modeling modules, resulting in a seamless transition from pose estimation to modeling.

Existing methods do not satisfy the requirements for on-the-fly global pose estimation. On the one hand, particle filters for mobile robot localization are able to keep pace with data acquisition. Unfortunately, they typically work either locally, are specialized to a certain sensor type, or cope only with 2D pose estimation. On the other hand, many global pose estimation methods cannot be adopted to work streamingly. Recently, we tried to fill this gap by introducing a particle filter registration [1] and adapting it to streaming pose estimation [2]. The work is aimed at applying in autonomous 3D modeling of unknown objects with laser stripers [3].

In this paper, we review these Monte Carlo methods and their performance. We focus especially on streaming registration, meaning that the pose is estimated on-the-fly during data acquisition. For autonomous object modeling, we propose to combine the registration method with the approach presented in [4] for creating complete object models. Further, we show various use cases, such as in robot assisted teleoperation, allowing for partial autonomy when grasping a power screwdriver as can be seen in Figure 1.

Compared to our previous works [1–3], new and extended experiments especially in autonomous 3D modeling and mobile robot localization are shown (Sections 8.3 and 8.4).



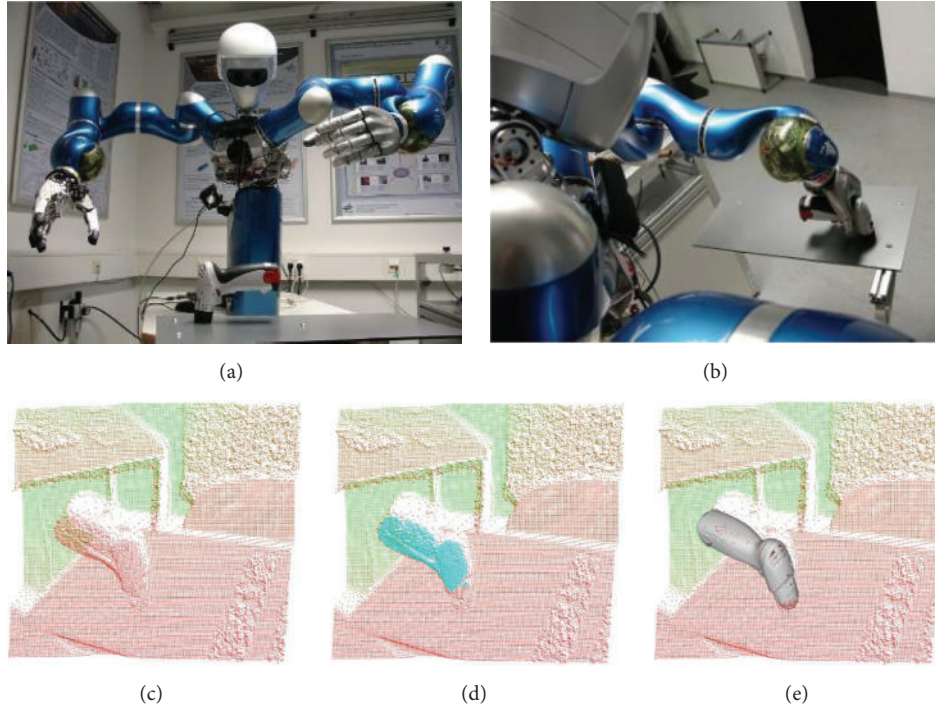


FIGURE 1: A humanoid robot (a) acquires a depth image of an object scene with a Time-of-Flight camera and registers the model of a power screwdriver ((c), (d), (e)) which allows for grasping the object (b). (c)–(e) depth values are color coded from red (close) to green (far), and the located model is shown in blue (points) and gray (CAD).

We enhance the streaming registration by providing a convergence criterion. Moreover, the influence of the sampling distribution, the introduced pose optimization during particle weighting, and the convergence criterion are investigated in detail. The main contributions in contrast to our previous works are

- (i) investigation of the influence of the proposed optimization step frequency on the robustness, reliability, and accuracy of results;
- (ii) investigation of the convergence behavior of the optimization frequency in comparison to the previous variants;
- (iii) definition of a convergence criterion;
- (iv) investigation of its influence on the robustness, reliability, and accuracy of results;
- (v) application in mobile robotics and comparison to a standard particle filter.

The remainder of this work is organized as follows. In Section 2, the related work concerning pose estimation and autonomous object modeling is reviewed. Then, in Section 3, the scalar features utilized in this work are introduced. Afterwards, particle filters are shortly reviewed and their application to registration is derived (Section 4). In the subsequent two sections, the sampling (Section 5) and scoring (Section 6) of particles are detailed. In Section 7, a feasibility study with the basic Monte Carlo method for global registration is performed. In Section 8, real experiments with

the streaming Monte Carlo registration and its results in the autonomous object modeling and mobile robot localization scenario are presented. The work is completed by a conclusion and a perspective on future work (Section 9).

## 2. Related Work

Registration is a crucial part of many technical applications such as reverse engineering, rapid prototyping, or manipulation tasks in manufacturing processes. It denotes the estimation of an unknown rigid motion between two 3D models of the same object.

Registration methods can be partitioned into fine (or local) and coarse (or global) ones. Fine methods require a good initial guess of the sought-after transformation. Coarse methods try to estimate the true rigid motion within a global search space. Both are reviewed in the following. For fine methods only a short overview including its application to mobile robot localization is given, as coarse methods are more closely related. Further, related work on real systems which autonomously model unknown objects is provided.

**2.1. Fine Registration Methods.** The iterative closest point (ICP) algorithm [5] is notably the most important representative of local methods (see [6] for an overview or [7] for a generalization). However, specialized methods have been proposed recently for RGB-D cameras [8–10] and image features [11, 12]. A further local technique related to our work is tracking with particle filters [13, 14], which is specialized for

fast local object movement estimation. Choi and Christensen [10] contribute a RGB-D object tracking approach implemented on a GPU and give an excellent general overview of particle filters in tracking problems, including edge based tracking, contour tracking, SLAM, and RGB-D based pose estimation. Therefore, the reader is referred to that work and the references cited therein for related tracking methods.

Localization and SLAM of mobile systems are also closely related but often specialized for local methods or methods restricted to 2D and could be categorized in between registration and tracking. A comprehensive overview of mobile robotics approaches is given by Sturm et al. [6]. Often particle filters based on pure depth images are used [15] which are not suitable for the stated problem, as the weighting is time consuming. Moreover, our estimation problem is not in 2D, and thus we need a much larger number of particles than classic mobile robotics particle filters. Additionally, we have to cope with higher update rates.

**2.2. Coarse Registration Methods.** In a large number of publications, contour and shape matching techniques are used, for which we refer the reader to the work of Ferrari et al. [16] and the literature cited therein. Note that we consider these techniques inappropriate for streaming pose estimation with laser stripers as they are slow and the contour or shape given by a few laser stripes is not very meaningful.

One of the most common methods in pose estimation is the random sampling consensus (RANSAC) introduced by Fischler and Bolles [17]. Its application to registration has been shown by Chen and Hung [18]. Commonly, subsets of points, point-normal pairs, or higher-dimensional features are sampled in the data sets to calculate unique rigid motions. Winkelbach [19] contributes an efficient way to sample point-normal pairs in order to build transformations. Drost et al. [20] also use point-normal pairs and a variant of the Generalized Hough Transform as voting scheme. Hillenbrand [21] contributes a robust cluster search in a set of transformations which are calculated from samples of either point-triples or point-normal pairs. Rusu et al. [22] use the Fast Point Feature Histogram in order to assign correspondences and use a sample consensus method in order to maximize the 3D overlap. Unfortunately, adapting these RANSAC-based methods to work with streaming data is not possible because a uniform sampling of points cannot be achieved before all data is acquired.

Another group of algorithms tries to group correspondences [23] or exploit salient points [24]. Aldoma et al. [23] evaluate various high-dimensional features for object recognition with a correspondence grouping method based on geometric consistency. A “center-star” variant of [24], followed by RANSAC-based filtering, yields a matching of similarly spaced point sets. In contrast to Rusu et al. [24] who search for salient points, points are sampled uniformly. The reduction to significant points is common. Gelfand et al. [25] reduce the data to a very small point set and apply a branch-and-bound algorithm to assign correspondences. The transformation is calculated by a least squares estimation. Cheng et al. [26] use a region growing algorithm to calculate

feature areas. In order to find correct correspondences, they use a relaxation labeling method. Both last methods rely on finding the unique and correct correspondences of feature points or areas. Again, this class of algorithms cannot be adapted to work with streaming data, since global data sets are needed, especially for feature calculation.

Barequet and Sharir [27] introduce a method that is based on the unique decomposability of rigid motions into a rotation and a translation. They iteratively search a discrete space of rotations by clustering the corresponding translations and finding the most definite cluster as the best rotation. In a later paper [28], they modify the method to work with directed features, that is, feature points with surface normals. They build  $\langle \text{feature point, normal} \rangle$  pairs to directly get possible rotations (with 1 free DOF).

A similar approach has been proposed by Tombari and Di Stefano [29]. The features they use yield a complete reference frame, not only a sole surface normal (as in [28]). Therefore, a correspondence pair defines a rigid motion, in contrast to a set of pure rotations. However, the scoring/voting table is the same (up to a constant translation) as in [28]. In order to calculate the best translation, Tombari and Di Stefano apply an ICP iteration to the correspondence pairs after voting. Barequet and Sharir use the found transformation directly (ICP can be applied to the whole data set afterwards). Tombari and Di Stefano also use their method for object detection and prove that it is more robust and reliable than other standard methods that use a pose space clustering or geometric consistency.

Glover et al. [30] use a Bayes filter for pose estimation, where the rotational part of the transformations is represented by a Bingham distribution and extend their approach for multiple object detection in cluttered scenes [31].

Rink et al. [1] reformulate Barequet and Sharir’s approach as a particle filter and show that, in applications with very noisy data, relying on accurate surface normals or reproducible reference frames (as in [29]) can fail. Thus, scalar feature descriptors are proposed. Furthermore, a comparison to similar strategies [22, 23, 27, 29] is given and the advantages of explicit integration of prior knowledge about the searched transformation are presented. In a subsequent paper Rink et al. [2] advance the idea of particle filtering with scalar features to streaming pose estimation, adapting the search space to the space of rigid body transformations and giving a theoretically sound weighting of particles. The streaming feature calculation in that approach is based on a streaming principal component analysis used for tangential plane estimation in streaming mesh construction, proposed originally by Bodenmüller [32]. In a recent paper, Rink and Kriegel [3] optimize their method for the application in autonomous 3D modeling. An optimization in the particle weighting step is introduced and sampling pose particles according to a truncated Bingham/normal distribution is compared to uniform sampling. Further, they integrate the streaming pose estimation into an autonomous 3D modeling approach.

**2.3. Autonomous Object Modeling.** In autonomous object modeling, usually a robot-sensor system is utilized to plan a Next-Best-View (NBV) in order to acquire a 3D model of

an unknown object. Although the area of NBV planning has been widely explored [33, 34], there is little research on real systems for autonomous object modeling.

For the purpose of modeling cultural heritage objects, Karaszewski et al. [35] present a measurement system comprising a turntable and a vertically moveable pedestal. First, they select areas in the boundary area as viewpoint candidates. Then, low point density areas are selected. The digitization time is several hours, even for small objects. Khalfaoui et al. [36] combine an industrial robot, a turntable, and a very large and expensive fringe projection system. In order to plan NBVs, they define barely visible surfaces as viewpoint candidates. They perform a mean shift clustering for NBV selection. In order to avoid viewpoints being very close to each other, they use a minimal distance criterion. In the resulting model, several holes remain. Torabi and Gupta [37] use a smaller robot with 2D range sensor and focus on the exploration, not on the object modeling. Vasquez-Gomez et al. [38] autonomously reconstruct unknown objects with a mobile manipulator and a Kinect sensor. In order to avoid collisions, NBVs are sampled in configuration space and evaluated in Cartesian space.

Kriegel et al. [4] combine an industrial robot with a laser striper and focus on high quality model acquisition. Therefore, they define a quality criterion and consider the surface quality during Next-Best-Scan (NBS) planning. None of these approaches are able to obtain the bottom part of an object. In [39] the last approach [4] is used to create a pose estimation data set. There, initially occluded parts are also modeled. However, the objects have to be repositioned manually about a defined axis quite perfectly because an ICP is used for registration.

**2.4. Distinction.** In this work, we present Monte Carlo registration methods and apply it to different scenarios. The streaming registration works with pure depth measurements and is thus not specialized for a particular sensor. Notably it also works with laser stripers. It is a global method and works streamingly. To the best of our knowledge, there is so far no method satisfying these requirements. In addition to the results presented in [1–3], we perform a more in-depth analysis of the theoretical basis for the presented method. Further, we review the Monte Carlo registration approaches with more details concerning the used sampling methods. We improve the original methods, especially by equipping the streaming pose estimation method with a convergence criterion and perform extended experiments. Moreover, we combine the registration method with the approach presented in [4] which focuses on the modeling and plans NBSs for a laser striper for creating complete object models. Thus, we extend the autonomous modeling system by enabling the acquisition of complete object models by arbitrary repositioning of objects, which has not been done so far.

### 3. Features

For a streaming calculation only local features can be used, since regional or global features are computationally too

expensive. In the literature, various multidimensional features exist [22] that work well with a low or moderate level of noise. In this work scalar features are used for two reasons. On the one hand, if a small neighborhood radius is used for feature calculation, a high level of noise increases the number of false matches [23]. This limits the advantage of the higher expressiveness over scalar features. On the other hand, with a large neighborhood radius, an iterative calculation is difficult to achieve because an exhaustive neighborhood search has to be performed. Moreover, the scalar curvature features used in this work already proved to be suitable for an iterative streaming calculation; see [32] for an application of some of the features as a quality measure for streaming surface normal estimation.

Point clouds and triangle meshes are common representations for 3D sensor data and can be directly computed from range images or from streams [32]. Examples for angular features working on different data types are given by Barequet and Sharir [27]. However, as special data structures are presumed there, the proposed mean angles for meshes cannot be used for the situation considered here. In the following we present some applicable features for homogeneous triangle meshes and point clouds and show how to compute them streamingly. Note that every feature point  $p = (c_p, n_p, v_p) \in \mathbb{R}^3 \times \mathcal{S}^2 \times \mathbb{R}$  comprises coordinates  $c_p$ , a surface normal  $n_p$  ( $\mathcal{S}^2$  being the unit sphere), and a feature value  $v_p$ .

**3.1. Normal Cosines in Polygon Meshes and Point Clouds.** Let in the following  $p$  be a point with surface normal  $n_p$  and neighborhood  $N(p)$ . For a  $q \in N(p) \setminus \{p\}$  we define

$$c(p, q) := \cos \left( n_p, \frac{q - p}{\|q - p\|} \right) \quad (1)$$

and call it normal cosine of  $p$  and  $q$ . Accordingly, the mean, maximum, and minimum of  $\{c(p, q) \mid q \in N(p) \setminus \{p\}\}$  are called the mean normal cosine (MNC), maximum normal cosine (MaNC), and minimum normal cosine (MiNC) in  $p$  with neighborhood  $N(p)$ , respectively.

In this work we use two types of neighborhoods for the calculation of features, depending on the data structure that is used. A polygon mesh contains a set of vertices  $\mathcal{V}$ , a set of edges  $\mathcal{E}$ , and a set of polygons. Each edge  $e \in \mathcal{E}$  is defined by two vertices  $v_1, v_2 \in \mathcal{V}$ , denoted by  $\langle v_1, v_2 \rangle$ . When dealing with homogeneous polygon meshes, it is reasonable to define the neighborhood of a vertex  $p \in \mathcal{V}$  as all points that are connected with  $p$  via  $l$  edges. It can be defined recursively by  $N_0(p) = \{p\}$  and

$$N_l(p) := N_{l-1} \cup \{q \in \mathcal{V} : \exists \tilde{p} \in N_{l-1}(p) : \langle \tilde{p}, q \rangle \in \mathcal{E}\}. \quad (2)$$

Figure 2 shows the MNC, the MaNC, and the MiNC of a triangle mesh of a wooden workpiece and their histograms. Note that points containing border vertices in their neighborhood are excluded from the feature calculation (depicted in white) because a robust feature calculation is not possible for this case. Since holes especially arise in high curvature areas,



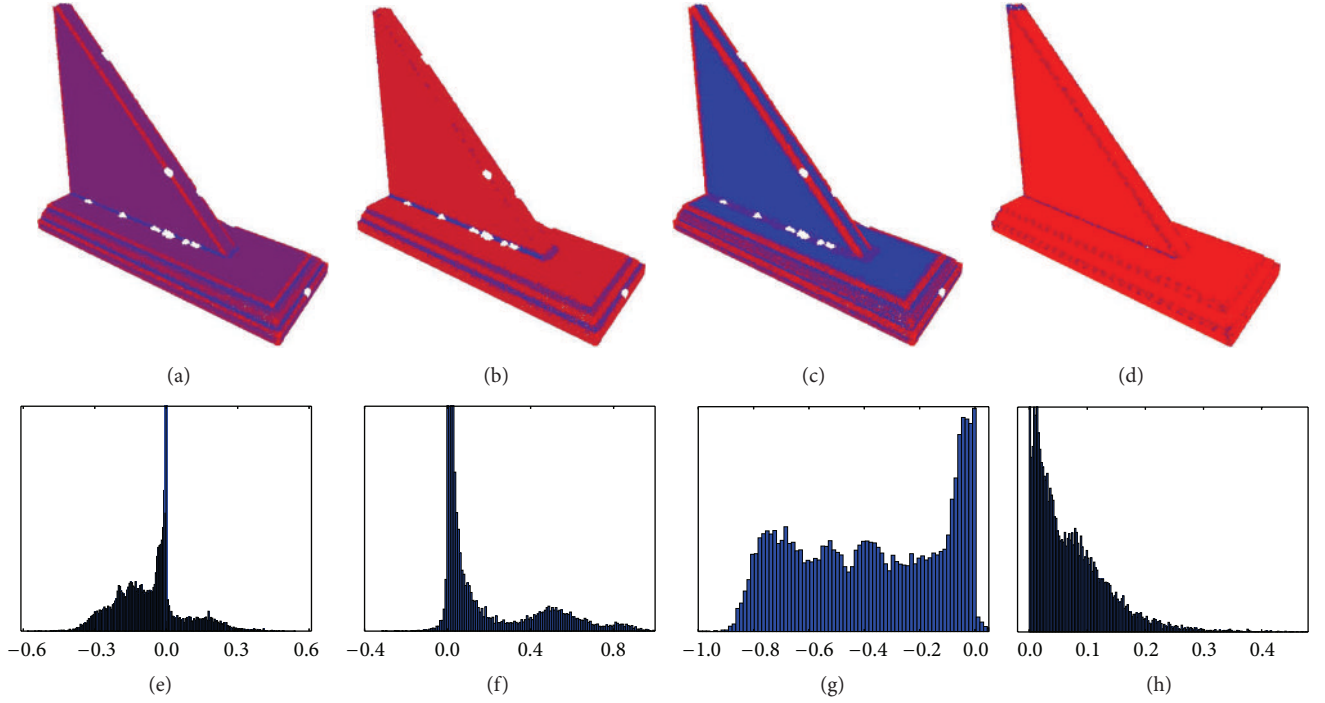


FIGURE 2: (a)–(d) The MNC, MaNC, MiNC, and EVQ13 of a wooden workpiece (high values are light red and low values are blue). (e)–(h) The corresponding histograms (most frequent value cut-off).

significant features could be incorrectly computed there. Note that in this case synthetic point data is used for mesh and feature generation. In nontechnical data sets, the distribution of the features is often closer to a normal distribution than in this example.

If point clouds are used or the polygon meshes are not homogeneous (i.e., edge lengths vary strongly), we use ball neighborhoods. For a point cloud  $P$  and a point  $p \in P$  let

$$N_r(p) := \{q \in P : \|p - q\| \leq r\}. \quad (3)$$

The normal cosine used in practice depends on the objects that are expected. Our experience is that convex regions in objects can be extracted with MNC and MiNC and concave regions with MNC and MaNC. For the general case we propose the MNC. If mainly convex regions are expected to be more discriminative we propose the MiNC and if mainly concave regions are expected we propose the MaNC.

**3.2. Point Clouds and Eigenvalues.** If no homogeneous polygon mesh or reliable normal estimation is given in advance, alternative features can be calculated. Let  $\lambda_1 \leq \lambda_2 \leq \lambda_3$  be the eigenvalues of a point neighborhood covariance matrix. Then  $\lambda_1/\lambda_3$  defines a curvature measure and can be used as geometric feature, denoted as eigenvalue quotient of eigenvalues 1 and 3 (EVQ13). The literature on similar features exists [40–43]. Figures 2(d) and 2(h) show the example features of a wooden workpiece. Note that missing features do not emerge, but ambiguity between convex and concave regions arises.

Flat objects like metal sheets do not yield relevant curvature features, disabling a robust pose estimation. In such cases we use the feature value  $\lambda_2/\lambda_3$ , denoted as EVQ23. This feature characterizes border points of the object (see Figure 13 for an example).

**3.3. Streaming Feature Calculation.** Feature-based streaming pose estimation requires streaming feature calculation which is implemented by a processing pipeline comprising three stages: the *density limitation*, the *normal estimation*, and the *feature generation* step. The depth points coming from a real-time data stream have to pass a limitation test in order to be inserted into the model: each newly acquired point that is closer than a distance  $r_r$  to any point already inserted into the model is rejected. Thus, the computational effort can be controlled because the entire Euclidean point density of the model is limited. For each point passing the *density limitation*, a surface normal is estimated using principal component analysis for all points within a ball neighborhood with radius  $r_n$ . Only points with a robust *normal estimation* (see [32] for details) are transferred to the subsequent *feature generation* step.

**3.3.1. Eigenvalues.** At the end of the *normal estimation* stage, the eigenvalues of the point neighborhood covariance matrix are readily available from the principal component analysis. Thus, the streaming feature calculation for EVQ13 and EVQ23 is straightforward: if a stable normal is ready, the corresponding feature point is calculated



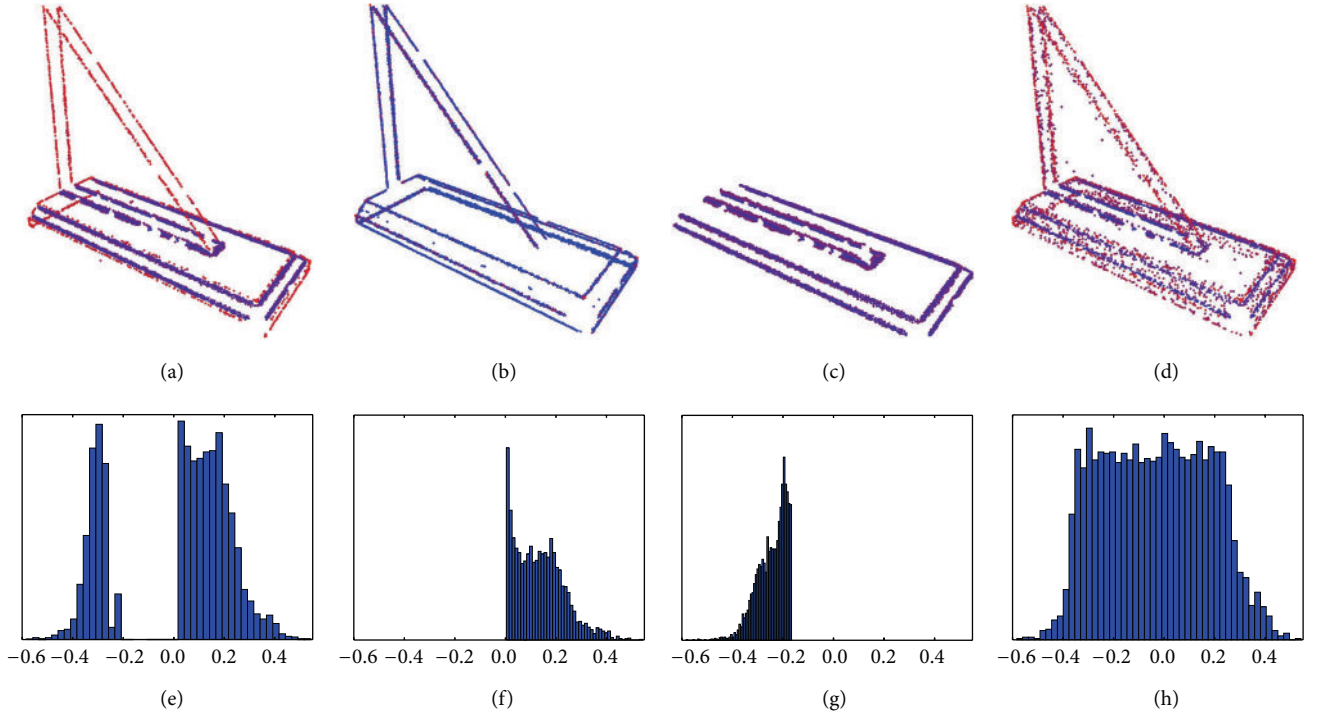


FIGURE 3: The reduction types (from left to right): removal of biggest bins, removal of leftmost bins, removal of rightmost bins, and randomized removal in biggest bins.

from the eigenvalues and inserted into the feature point stream.

**3.3.2. Normal Cosines.** The proposed angle features are calculated from the point surface normal and the neighborhood points in the model. Consequently, if a stable normal is ready and the new point is inserted into the module, the MNC, MaNC, or MiNC is calculated in the neighborhood of radius  $r_n$  (immediately available from the *normal estimation* stage). Also, all the points in the neighborhood are updated correspondingly.

**3.3.3. Automatic Feature Type Selection.** Theoretically, the most appropriate feature can be automatically selected by calculating the MNC and evaluating the histogram. For unimodal symmetric distributions, the MNC should then be used. If the distribution is skewed, the MiNC could be used for a heavy left tail and the MaNC for a heavy right tail. In practice, however, the automatic choice of the other parameters (like the neighborhood radius or meshing parameters) used for feature calculation poses a problem, especially if computation time is an issue.

**3.4. Feature Point Reduction, Classification, and Correspondences.** Besides the correspondence assignment features can be used to reduce the data to significant points. On the one hand this leads to lower computational costs. On the other hand correspondence assignment is concentrated on

characteristic areas of the object, reducing the number of false positive correspondences.

Therefore, the feature values are first summarized in a histogram. Then the histogram's bins can be removed iteratively. Either the biggest bin (see Figure 3) or the leftmost/rightmost bin is removed until only a given percentage of the original data remains. Alternatively, it can be useful to keep some points from the biggest histogram bins, for example, when dealing with great flat areas. Then, the smallest bins are kept and from the remaining bins a certain fraction is sampled so that the desired overall fraction remains in the end. Figure 3 shows the differences in these reduction strategies for the MNC. In ((a), (b), (c), (d)), the remaining feature points are depicted. In ((e), (f), (g), (h)), the corresponding histograms of feature values are shown. Clearly, the removal of the biggest bins results in remaining convex and concave extrema. The removal of the leftmost/rightmost bins results in remaining convex/concave extrema. In this case, weighted random reduction in the biggest bins results in evenly sized bins for the most frequent feature values.

In order to assign correspondences, we use discretized feature values, so-called feature classes, conferring some robustness to noisy sensor data. Given a number  $n$  of feature classes, the classes are defined uniquely by the maximum feature  $f_{\max}$ , the minimum feature  $f_{\min}$ , and an equal bin width  $b = (f_{\max} - f_{\min})/n$ . In this work, values of  $n = 7$ ,  $n = 5$ , or  $n = 3$  are used. Then, every feature point of one data set corresponds to every feature point with a feature of the same class in the other data set. This implicates a normalization and allows the data sets to be measured with different sensors.



FIGURE 4: The noise in a ToF cam depth image prevents reliable geometric feature calculation. The power screwdriver is colored light blue.

**3.5. Features in Time-of-Flight Camera Data.** Robust feature calculation from noisy Time-of-Flight camera (ToF) data is a special challenge. Figure 4 illustrates the noise with an exemplary depth image of a power screwdriver. In such cases, the selection of key points relying purely on geometry fails. Though, we take advantage of the additional intensity image provided by ToF cams because optical and geometric edges often coincide in the data of technical products. In this particular use case, we search features on the handle of the screwdriver. On the CAD data template, the fine geometric features can easily be detected; see Figure 5. In the real ToF cam depth image, these geometric edges cannot be found. However, the intensity gradients in the intensity image include the searched edge at the handle of the power screwdriver. As they also include the transition between the shape of the screwdriver and the background, edges in the range image must be subtracted. Therefore, we use an image filter combination consisting of dilation, blurring, and a Canny edge detector on the intensity image as well as on the depth image to acquire the edges (for filtering the software OpenCV 2.4 is used). Finally, the depth edge image is subtracted from the intensity edge image, as shown in Figure 6. In total, exactly the geometric edges we sought for are extracted. Section 7.2.2 shows an application and more details.

## 4. Monte Carlo Registration

In this section, we introduce the basic Monte Carlo registration methods. First, we review the general particle filter and its specialization for registration. Then, the proposed methods are derived.

**4.1. Particle Filter.** For a detailed description of particle filters and their applications the reader is referred to the literature [10, 13–15]. Here, just the notations used in subsequent sections are given and the most important points are reviewed.

Let  $X$  be a random variable and let  $f(X | \theta)$  be the probability density function (pdf) of  $X$  conditioned on some parameter  $\theta$ . Let further  $p(\theta)$  be the a priori pdf of the parameter  $\theta$ . According to Bayes' rule, after some observation  $x$  of  $X$ , the posterior of  $\theta$  is given by

$$p(\theta | x) \propto f(x | \theta) p(\theta), \quad (4)$$

with  $\propto$  being the symbol for proportionality. If repeated measurements of  $X$  are carried out, Bayes' rule can be applied iteratively (if the measurements are statistically independent). Suppose that the state of the parameter  $\theta$  changes between the observations  $x_i$  and  $x_{i+1}$  of  $X$  according to a transition function  $A_i$ :

$$\theta_{i+1} = A_i(\theta_i) + \varepsilon_{A_i}, \quad (5)$$

where  $A_i(\cdot)$  describes a systematic change with an error  $\varepsilon_{A_i}$  that follows the pdf  $g_{A_i}$ . The transition  $A_i$  is changing over time in many applications. Moreover,  $\varepsilon_{A_i}$  often depends on  $A_i$ . Therefore, the pdf  $g_{A_i}$  is changing, too. In the case of registration,  $A_i$  as well as  $\varepsilon_{A_i}$  can be assumed to be constant. The assumption of first-order Markov chains and the independence of  $\theta_{i+1}$  from  $x_i$  yields

$$\begin{aligned} p(\theta_{i+1} | x_{i+1}, \dots, x_1, \theta_i, \dots, \theta_1, A_i) \\ &= p(\theta_{i+1} | x_{i+1}, \theta_i, A_i) \\ &\propto f(x_{i+1} | \theta_{i+1}, \theta_i) p(\theta_{i+1} | \theta_i, A_i) \\ &= f(x_{i+1} | \theta_{i+1}) p(\theta_{i+1} | \theta_i, A_i). \end{aligned} \quad (6)$$

Now let the pdf of  $\theta_i$  be represented by  $m_i$  particles:

$$(\theta_i^j, w_i^j)_{j=1, \dots, m_i} \quad (7)$$

with sampled states  $\theta_i^j$  and corresponding weights  $w_i^j$ . After a transition according to  $A_i$  and a new observation of  $X$ , new particles are sampled according to (5). Note that sampling from the pdf  $g(\varepsilon_{A_i})$  has to be possible, which is not generally the case. Therefore,  $\varepsilon_{A_i}$  is typically assumed to be distributed normally or uniformly. The new particles are each weighted with

$$w_{i+1}^j = f(x_{i+1} | \theta_{i+1}^j), \quad j = 1, \dots, m_i. \quad (8)$$

Finally,  $m_{i+1}$  particles are resampled according to these weights. Thus, particle filtering can be summarized as follows:

- (1) Initialize ( $i = 0$ ) the  $m_0$  particles  $\theta_0^j$ .
- (2) Collect data  $x_i$  and weight the particles with

$$w_i^j := f(x_i^j | \theta_i^j). \quad (9)$$

- (3) Resample  $m_{i+1}$  particles with the weights  $w_i^j$ .
- (4) Apply the transition by  $\theta_{i+1}^j = A_i(\theta_i^j)$ .
- (5) Set  $i := i + 1$  and return to Step (2).

**4.2. Monte Carlo Registration.** In the case of Monte Carlo registration, the particles can either be sampled in the space of rotations [1] or rigid body transformations [2]. Let  $\mathcal{T}$  be the corresponding state space in either case. Further,  $T_i \in \mathcal{T}$  denotes the unknown transformation between two models  $P$  and  $Q$  of the same object at time step  $i$ . Then, every particle

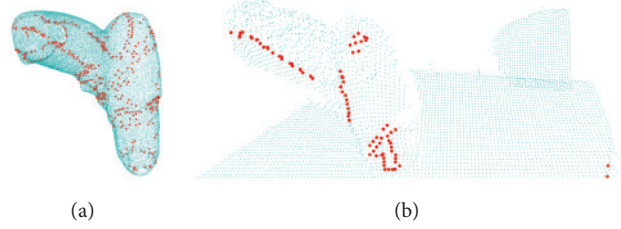


FIGURE 5: Point cloud models describing a screwdriver: (a) shows the template point cloud model (light blue) and the calculated feature points (red). (b) shows the cropped depth image with emphasized feature points (red) extracted from the ToF data (before plane deletion).



FIGURE 6: Edges detected in cropped ToF data. From (a) to (d) intensity, intensity edges, range edges, and edge difference image. In the intensity image the color gradient of the handle of the power screwdriver being used as a feature is conspicuous. This gradient can be recognized in the (d) difference image again.

comprises a transformation  $T \in \mathcal{T}$  and a weight  $w$ . The state transition between two time steps is the identity in the case of registration because the observed object is not moving ( $A_i = \text{id}$ ). The distribution of the error  $\varepsilon_{A_i}$  can be assumed to be a uniform or normal distribution. In each time step  $i$ , all particles  $(T, w)$  are weighted by the pdf  $f(Q_i | T, P)$  of observed data  $Q_i$  conditional on the state  $T$  and the template model  $P$ . Then, particle filter registration can be summarized as follows:

- (1) sample pose particles initially;
- (2) weigh particles by  $f(Q_i | T, P)$ ;
- (3) resample particles according to their weights;
- (4) optionally: adapt some parameters;
- (5) sample particles in neighborhoods of existing particles; return to Step (2) if not converged.

**4.3. MCR and SMCR.** The Monte Carlo registration method presented in [1] is denoted by MCR in this paper. It works offline and uses all feature points in every weighting step and the particles are rotations. The particle weighting is done by a heuristic cluster evaluation.

The streaming Monte Carlo registration method presented in [2] is denoted by SMCR in this paper. It works streamingly and can use either new incoming feature points or all accumulated feature points in the weighting steps. The particles are rigid body transformations. The particle weighting is done with respect to a truncated normal distribution. For SMCR, the number of particles, the sampling radii, and the radius for the score function from Section 6 are adapted in each Step (4) of Section 4.2. Each of these parameters has a maximum value and is reduced by a factor of 0.8 in Step (4), until a minimum value is reached.

## 5. Sampling Pose Particles

Initially the particles are sampled uniformly in a region according to the prior knowledge. In the subsequent iterations only neighborhood sampling is performed. This is done either with truncated uniform distributions or truncated normal/Bingham distributions. The radius of the truncated neighborhood can be adapted with convergence over time.

As sampling of translations uniformly or normally in neighborhoods is trivial, only the sampling of rotations is detailed here. Rigid body transformations are sampled by sampling the rotational and translational parts separately.

**5.1. Sampling Unit Vectors Uniformly.** A prerequisite for our approach to sample rotations uniformly is to sample uniformly distributed points in neighborhoods on the unit sphere  $\mathcal{S}^2$  in  $\mathbb{R}^3$ . Sampling a point on the unit sphere can be achieved by the following steps:

- (1) Sample  $v$  uniformly distributed in  $[-1; 1]$ .
- (2) Sample  $u$  uniformly distributed in  $[-\pi; \pi]$ .
- (3) Calculate  $\mathbf{c} := \cos(\arcsin(v))$ .
- (4) Build  $p = (\mathbf{c} \cdot \cos(u), \mathbf{c} \cdot \sin(u), v)^T$ .

According to the transformation theorem for densities,  $p$  will be uniformly distributed on the unit sphere [44].

Let the  $\alpha$ -neighborhood  $N_\alpha(a)$  of a vector  $a \in \mathcal{S}^2$  be defined as

$$N_\alpha(a) := \{\tilde{a} \in \mathcal{S}^2 \mid \angle(a, \tilde{a}) \leq \alpha\}, \quad (10)$$

where  $\alpha \in [0, \pi]$ .

Then, the above approach can be adapted to sample unit vectors in an  $\alpha$ -neighborhood of the first standard basis

vector  $e_1$  with  $\alpha \leq \pi/2$ . Just modify steps one and two as follows:

- (1) Sample  $v$  uniformly distributed in  $[-\sin(\alpha); \sin(\alpha)]$ .
- (2) Sample  $u$  uniformly distributed in  $[-\alpha; \alpha]$ .

This specialization is biased for  $\alpha > \pi/2$ . In this case simple rejection sampling can be used.

**5.2. Sampling Rotations in Neighborhoods Uniformly.** In statistics, various strategies are common to choose priors that express ignorance about a parameter. In this work, uniform distributions are used for the initial sampling of transformations. The most common way to achieve a deterministic uniform sampling in a space is to build a homogeneous grid in that space. However, this leads to biased grids when dealing with the common representations of rotations. Sampling matrices straightforward in this manner is not even possible. There are sophisticated algorithms for deterministic uniform sampling of rotations [45–47], but unbiased deterministic sampling is not possible.

However, uniform sampling in a statistical sense can be achieved [48, 49] and is advantageous compared to simple grid sampling on Euler angles [1]. In the remainder of this section we detail our variant of Shoemake's method [49] to sample rotations uniformly in neighborhoods. Let  $\mathcal{R}$  be the space of rotations in the remainder. The  $\alpha$ -neighborhood  $N_\alpha(R)$  of a rotation  $R$  is defined as

$$N_\alpha(R) := \{\tilde{R} \in \mathcal{R} \mid d(R, \tilde{R}) \leq \alpha\}, \quad (11)$$

where  $\alpha \in [0, \pi]$

with  $d(R, \tilde{R})$  being the rotational difference between two rotations  $R, \tilde{R}$ , that is, the absolute value of the angle of the axis-angle representation of  $R \circ \tilde{R}^{-1}$ . Let  $e_1, e_2, e_3$  be the basis vectors of the base coordinate system. In order to sample rotations on  $\alpha$ -neighborhoods, we represent a rotation by the transformed coordinate system  $\tilde{e}_1, \tilde{e}_2, \tilde{e}_3$ , that is, the columns of the rotation matrix. Sampling rotations in a neighborhood  $N_\alpha(R)$  of  $R$  is done by sampling a rotation  $R_\alpha$  in  $N_\alpha(\text{id})$  and calculating  $\tilde{R} = R \circ R_\alpha$ . In order to sample in  $N_\alpha(\text{id})$ , we propose the following procedure (see Figure 7):

- (1) Sample  $\tilde{e}_1$  in the  $\alpha$ -neighborhood of  $e_1$ .
- (2) Calculate the vector

$$v_3 = \begin{cases} e_3, & \text{if } \tilde{e}_1 = \pm e_2, \\ \tilde{e}_1 \times e_2, & \text{else.} \end{cases} \quad (12)$$

- (3) Calculate the vector  $v_2 = v_3 \times \tilde{e}_1$ .
- (4) Rotate vectors  $v_2, v_3$  around  $\tilde{e}_1$  with a random angle  $\varphi$  onto  $\tilde{e}_2$  and  $\tilde{e}_3$ . If  $\alpha < \pi/2$  sample  $\varphi$  in  $[-\alpha; \alpha]$ . Otherwise, sample  $\varphi$  in  $[-\pi; \pi]$ .
- (5) Build the rotation matrix  $R := (\tilde{e}_1, \tilde{e}_2, \tilde{e}_3)$ . If the rotation angle holds  $d(R, \text{id}) \leq \alpha$  accept it; else return to Step (1).

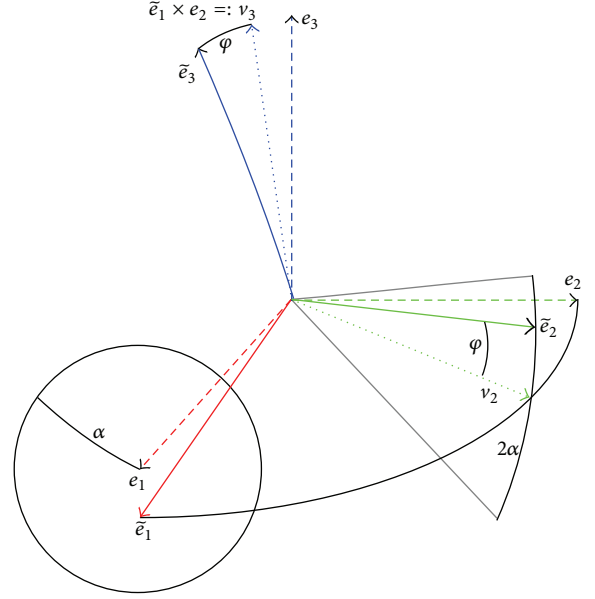


FIGURE 7: Uniform neighborhood sampling of rotations:  $\tilde{e}_1$  is sampled uniformly in the  $\alpha$ -neighborhood of  $e_1$ . Then  $v_2$  is chosen perpendicular to  $\tilde{e}_1$  and in the plane spanned by  $\tilde{e}_1$  and  $e_2$ . Rotating around  $\tilde{e}_1$  with a random angle smaller than  $\alpha$  makes  $v_2$  independent of  $\tilde{e}_1$ .

Figure 7 illustrates these relations:  $v_2$  is perpendicular to  $v_3$ , which in turn is perpendicular to  $\tilde{e}_1$  and  $e_2$ . Therefore,  $v_2$  lies in the plane spanned by  $\tilde{e}_1$  and  $e_2$  and thus  $v_2$  is as close as possible to  $e_2$  conditioned on a fixed  $\tilde{e}_1$ . This assures an overall rotation angle as small as possible before applying the random rotation around  $\tilde{e}_1$  in Step (4). Rotating around  $\tilde{e}_1$  moves  $v_2$  onto  $\tilde{e}_2$ . For  $\alpha < \pi/2$ , rotations around  $\tilde{e}_1$  by an angle greater than  $\alpha$  cause  $v_2$  to leave the  $\alpha$ -neighborhood of  $e_2$ . For  $\alpha \geq \pi/2$ , this is not necessarily the case. Together, this motivates Step (4) because each transformed basis vector  $\tilde{e}_i$  must not lie outside the  $\alpha$ -neighborhood of the basis vector  $e_i$  if the resulting rotation is to lie in the  $\alpha$ -neighborhood of the identity. Rotations that lie outside the neighborhood are removed by Step (5). Figure 8(a) shows uniformly sampled rotations in an  $\alpha$ -neighborhood with  $\alpha = 90^\circ$ . The rotations are represented by the transformed unit vectors  $\tilde{e}_1$  and  $\tilde{e}_3$ , that is, the first and last columns of the corresponding rotation matrix. For better visibility,  $e_2$  was left out.

**5.3. Truncated Bingham Sampling.** On the space of rotations, no normal distribution exists, though so-called projected Gaussians have been proposed [50]. Most similar are special cases of Bingham distributions, which have already been used for pose estimation [30]. We propose to sample from a truncated special case of a Bingham distribution, with rejection sampling. Shortly, after sampling a rotation  $R$  uniformly in an  $\alpha$ -neighborhood of a rotation  $\tilde{R}$ , we do rejection sampling with the weight  $\exp(d(R, \tilde{R})^2/2\sigma^2)$ . This sampling strategy is easy to implement and if  $\sigma^2$  is chosen appropriately (independent of  $\alpha$ ), the computation time is negligible compared to the weighting step of the particles.



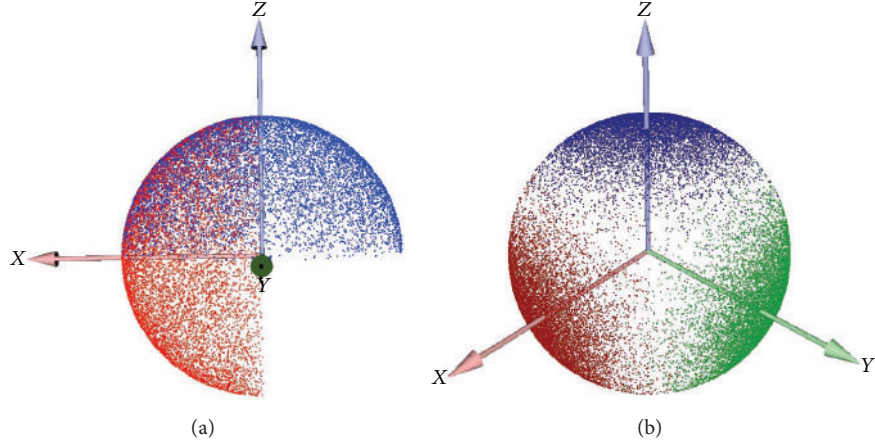


FIGURE 8: Uniformly (a) and truncated Bingham (b) distributed rotations:  $10^5$  images  $R(e_1)$  (red),  $R(e_2)$  (green, removed in (a) for better visibility), and  $R(e_3)$  (blue) with neighborhood radius  $\alpha = 90^\circ$  from different views. Variance of the normal is  $\sigma^2 = (18^\circ)^2$ .

Figure 8(b) shows truncated Bingham sampled rotations in a neighborhood of radius  $90^\circ$ . For better visibility, the normal distribution's variance is chosen as  $18^\circ$  in this example. In practice, we choose the standard deviation to be half the neighborhood radius.

## 6. Scoring Pose Particles

In this section, the scoring methods for rotations and rigid body transformations are presented. Scoring of a rotation is done by a cluster evaluation on the set of corresponding possible translations. This voting scheme is very similar to the generalized Hough transform. Scoring of a rigid body transformation is done by evaluating a truncated normal pdf. The former voting scheme is used in MCR and the latter in SMCR.

**6.1. Scoring Rotations.** In order to score a rotation, it is first applied to the corresponding data set. Then, all translational differences between all corresponding points are calculated (correspondences between the data sets are defined by equal feature classes). These differences define the set of all possible translations for the considered rotation. In order to find the one rotation with the most clustered set of such translations, two strategies are considered in this work. The first is to store the translations in a three-dimensional voting table and use the maximum number of elements in one bin as score. This method will be denoted in table in the remainder and is detailed in [27–29]. The second is to store the translations in a voxel space and use the maximum number of neighbors in a ball neighborhood as score; see Figure 9. This scoring method will be denoted as nb and was originally presented in [1]. The pdf  $f(Q_i | R, P)$  (see Section 4.2) cannot be used as score function, since it is not known and there is no reasonable assumption about it.

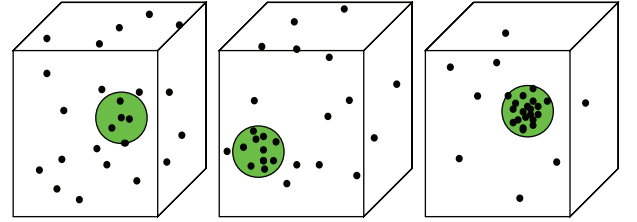


FIGURE 9: Scoring rotations: the maximum number of translations in a ball neighborhood in the set of translations defines a score for the clusteredness.

**6.2. Scoring Rigid Body Transformations.** The scores of the particles representing rigid body transformations are calculated according to a truncated normal distribution of the feature point coordinates, conditional on the pose. Let  $P, Q$  be the feature point sets of the template and the incoming data, correspondingly. Further, let the features be classified; that is,  $v_p$  is a discrete category for every  $p \in P$  (and correspondingly for  $Q$ ). Every particle describes a rigid body transformation  $T = (R, t)$ , defined by a rotation  $R$  and a translation  $t$ . In the following,  $q \in Q$  is corresponding to  $p \in P$ . For a known transformation  $T$  between data and template, it is reasonable to assume that  $c_q$  follows a normal distribution with expectation  $T(c_p)$  and a covariance matrix  $\Sigma = \sigma^2 \cdot \text{id}$ . If the errors are identically and independently distributed and a set of feature points  $\mathbf{p} = \{p_1, \dots, p_n\}$  and a set of correspondences  $\mathbf{q} = \{q_1, \dots, q_n\}$  are given, the conditional pdf of all feature point locations is

$$f(\mathbf{q} | T, \mathbf{p}) \propto \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^n \|T(c_{p_i}) - c_{q_i}\|^2 \right). \quad (13)$$

The corresponding  $p_i$  are approximated by the nearest feature point to  $q_i$  with the same feature class:

$$p_i := \arg \min_{p \in P, v_p = v_{q_i}} \|T(c_p), c_{q_i}\|. \quad (14)$$



For a  $q$  with erroneous feature class, no correct corresponding  $p$  will be found in the template. The best we can do in this case is to assume a uniform distribution of the feature point location, conditional on a wrong corresponding  $p$ . For some distance threshold  $r_{\max}$  we define

$$d_i := \min \{ \|T(c_{p_i}) - c_{q_i}\|, r_{\max} \} \quad (15)$$

and adopt (13) to

$$f(\mathbf{q} | T, \mathbf{p}) \propto \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^n d_i^2 \right) \quad (16)$$

which corresponds to (truncated) normally distributed errors if the correspondences are found within a radius  $r_{\max}$  and uniformly distributed errors if not (with the density equal to that of the truncated normal at its boundary). This is actually an improper pdf because its integral is unbounded. Nevertheless, sampling importance resampling [51] is possible with it. Thus, each transformation  $T$  is scored with feature points  $Q$  as follows. First, each point  $q_i \in Q$  is classified according to the class borders of the template. It is transformed to  $c_{q_i} := T^{-1}(c_{q_i})$  and the template model is searched for the nearest feature point  $p_i$  of the same feature class. The corresponding distance is denoted as  $d_i := \|c_{q_i} - c_{p_i}\|$ . If no such point is found within the search radius  $r_{\max}$ , the distance is set to this radius ( $d_i := r_{\max}$ ). Based on the distances  $d_i$  of all available feature points  $p_i$ , the particle weight is defined by

$$w(T) := \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^n d_i^2 \right). \quad (17)$$

**6.2.1. Scoring Variants.** Theoretically, a particle filter uses only statistically independent measurements for each update; that is, only new incoming feature points are utilized for  $Q$ . Let therefore  $Q_j$  be the incoming set of feature points at time step  $j$ . The results in Section 8.2 will show that  $Q = Q_j$  leads to poor convergence behavior. The convergence can be improved by using all previously measured feature points for  $Q$ ; that is,  $Q := \bigcup_{k=0}^j Q_k$ . In order to distinguish the two scoring variants, we call the former *streaming particle filter registration* (SPFR) and the latter *streaming Monte Carlo registration* (SMCR) in the remainder. As shown in [2, 3] and reviewed in Section 8.2, SMCR has a better convergence behavior. Note that the offline variant MCR for global registration always uses all feature points.

**6.2.2. Optimization.** If a  $p_i$  with  $d_i \leq r_{\max}$  according to (14) and (15) is found, it defines a correspondence to  $q_i$ . Thus, each particle yields a set of correspondences in the weighting step. In order to correct the pose particle, these correspondences can be used for an ICP iteration. If such an optimization step is applied, the method is denoted by SMCRO in the remainder. An additional subscript defines the frequency of such an optimization; for example, SMCRO<sub>5</sub> denotes an optimization in every fifth update.



FIGURE 10: The two data sets obtained by selecting parts of the Stanford Bunny.

**6.2.3. Convergence Criterion.** One advantage of streaming pose estimation is the possibility of stopping data acquisition as soon as the estimation converged. For this purpose a convergence criterion for SMCR is introduced. In every update step we calculate the rotational and translational difference of the highest rated transformation to the highest rated transformation from the last step. If the differences are below some thresholds  $c_r$  and  $c_t$  in five consecutive steps, the calculation is aborted. We combine the optimization in every step with this criterion and denote the method by SMCRC.

## 7. Feasibility Study with MCR

In this section, we briefly summarize the most important results that are obtained with MCR. For a more detailed review, the reader is referred to [1]. The results serve as feasibility study on the question, whether this kind of particle filter can compete with the state-of-the-art registration methods. Otherwise, the effort of developing a streaming variant would not be justified.

**7.1. Validation with Artificial Data.** For the validation with artificial data, two submeshes of the well-known Stanford Bunny are extracted; see Figure 10. The computing time  $\bar{t}$ , the mean rotational error  $\bar{\rho}$ , and the percentage of successful estimations  $sr$  are investigated. A successful estimation is defined by a rotational error less than  $20^\circ$  and reflects robustness as such errors can be equalized by ICP. All experiments in this section were performed by 1000 test runs. In each of these test runs, the underlying rotation was chosen randomly.

First, three different sampling strategies are examined. The first is the original one [27], denoted by *det* in the following. There, rotations are represented by Euler angles and sampled initially on a homogeneous grid with resolution  $\rho_0$ . In each further step only the best rotation is kept and neighboring rotations are resampled on a local grid with 27 grid points, including the currently best rotation as center point. Subsequently, two stages are carried out: a coarse neighborhood search and a fine neighborhood search. The initial grid resolutions of the coarse and fine search are denoted by  $\rho_1$  and  $\rho_2$  in the following. Both in the coarse and in the fine search the grid resolution is adapted depending on whether a better rotation has been found in the last step or not. If a better rotation has been found in the coarse search, the local grid sampling is repeated around that rotation with the initial coarse resolution  $\rho_1$ ; otherwise the local search

TABLE 1: Mean computing time  $\bar{t}$ , mean rotational error  $\bar{\rho}$  in degree, and number of successful estimations in percent  $n(\rho)$ . Entries are for sampling types det/randdet/rand.

$(\rho_0, \rho_1, \rho_2)$	(5, 3, 1)	(20, 10, 5)	(50, 20, 10)
$\bar{t}$ [s]	<b>102/134/179</b>	10/9/11	1.7/1.8/ <b>0.5</b>
$\bar{\rho}$ [deg]	77/43/ <b>11</b>	15/13/7	73/ <b>63</b> /88
sr [%]	45/54/ <b>95</b>	93/95.4/ <b>98.6</b>	60/ <b>64.5</b> /20

is repeated with with a doubled resolution. If a maximum resolution is exceeded, the coarse search is aborted. Then, the fine search is started with the initial resolution  $\rho_2$ . If a better rotation can be found, the local grid sampling is repeated around that rotation with the initial fine resolution  $\rho_2$ ; otherwise the current resolution is cut in half and the local search is repeated. If the resolution falls below a minimum, the search is finished.

The second sampling strategy (denoted by randdet) is a mixture of discrete and random sampling: the initial samples are drawn as before. The neighborhood search is performed by randomly sampling 27 new rotations in the neighborhood (with adapted radius as before) of the best rotation of the previous step.

The third method is denoted by rand and is an important resampling approach: in every step samples are drawn randomly and resampled according to the scores. Therefore, not only in the neighborhood of the best rotation new samples are drawn, but in the neighborhood of all rotations. In order to assure comparability, initially the same number of rotations is sampled as in the first method. This number is defined by the initial resolution  $\rho_0$ . In each further step, the number of samples and the neighborhood radius are reduced by a factor of 0.5.

As convergence to the correct rotation is assumed, the resolution in the scoring (the bin width of table or the ball radius of nb in Section 6.1) is also adapted for each sampling stage. Sampling with det comprises three stages: an initial, a coarse, and a fine search. The bin widths of table in these stages are denoted as  $\tau_0, \tau_1, \tau_2$ , respectively, and are decreasing:  $\tau_0 > \tau_1 > \tau_2$ .

The results concerning sampling and scoring strategies are depicted in Tables 1 and 2 and can be summarized as follows: if the particle number is chosen properly, the proposed sampling rand outperforms the original method det concerning robustness and accuracy. The proposed scoring method nb yields slightly better results in accuracy and robustness compared to the original table. Though a computationally expensive neighborhood search in an octree has to be performed and thus computation time is much higher compared to computing a discretized 3D vote. Therefore, nb can only be recommended if computation time is irrelevant. Therefore, table and rand are used in the remainder.

Finally, a comparison of MCR to alternative approaches implemented in PCL was performed. These are the Hough voting method of Tombari and Di Stefano [29], the Geometric Consistency (GC) approach used by Aldoma et al. [23], and the SAC-IA method of Rusu et al. [22]. Figure 11 shows

TABLE 2: Mean computing time  $\bar{t}$ , mean rotational error  $\bar{\rho}$  in degree, and number of successful estimations in percent. Entries are for scoring methods table and nb.

$(\tau_0, \tau_1, \tau_2)$	(1, 0.8, 0.5)	(3, 2, 1)	(5, 3, 1)
$\bar{t}$ [s]	<b>10/60</b>	5/41	5/69
$\bar{\rho}$ [deg]	<b>53/56</b>	47/35	39/ <b>36</b>
sr [%]	61/ <b>63</b>	65/ <b>73</b>	69/ <b>73</b>

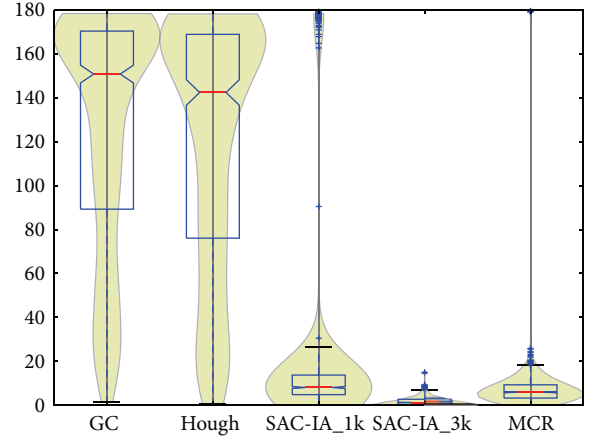


FIGURE 11: The rotational errors for the Stanford Bunny, comparison with methods available in PCL. A standard boxplot from python's matplotlib-package is overlaid with a density plot, in order to capture the multimodal distribution of the errors.

violin plots [52] of the results, including two versions of SAC-IA, first with the default 1000 (1k) iterations, then with an increased number of 3000 (3k) iterations. Notably, SAC-IA always runs up to the maximum allowed number (1000 or 3000 in this test). It maximizes the overlap quality directly and therefore performs well with low-dimensional features that are not as descriptive. Typically an increased number of iterations is needed to outperform MCR. MCR and SAC-IA clearly outperformed both GC and Hough when using the scalar features.

**7.2. Experiments with Real Data.** In the following, selected results with real data from different 3D sensors are presented, showing the effectiveness of MCR under hard conditions like small overlap areas and noisy data.

**7.2.1. Registration with Laser Striper.** In the first experiments, we use the DLR Multisensory 3D Modeler [53] that comprises a laser-line projector and a stereocamera system, implementing a laser-stripe range sensor. It is attached to a 6 DOF industrial robot, the KUKA KRI6-2.

**Registration of a Wooden Workpiece.** Figures 12(a), 12(b), 12(c), and 12(d) show the feature points and the reduced feature points used to match two scans of a wooden workpiece. Figures 12(e) and 12(f) show the result: the complete surface

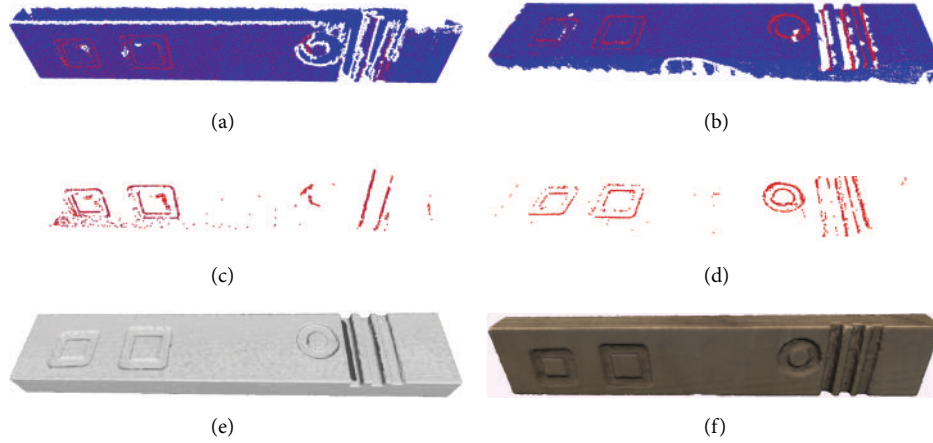


FIGURE 12: Modeling a wooden workpiece. Two sets of feature points ((a) and (b)), the reduced feature points ((c) and (d)), and the result after registration ((e) and (f)): the remeshed model (e) and the whole model with textures (f).

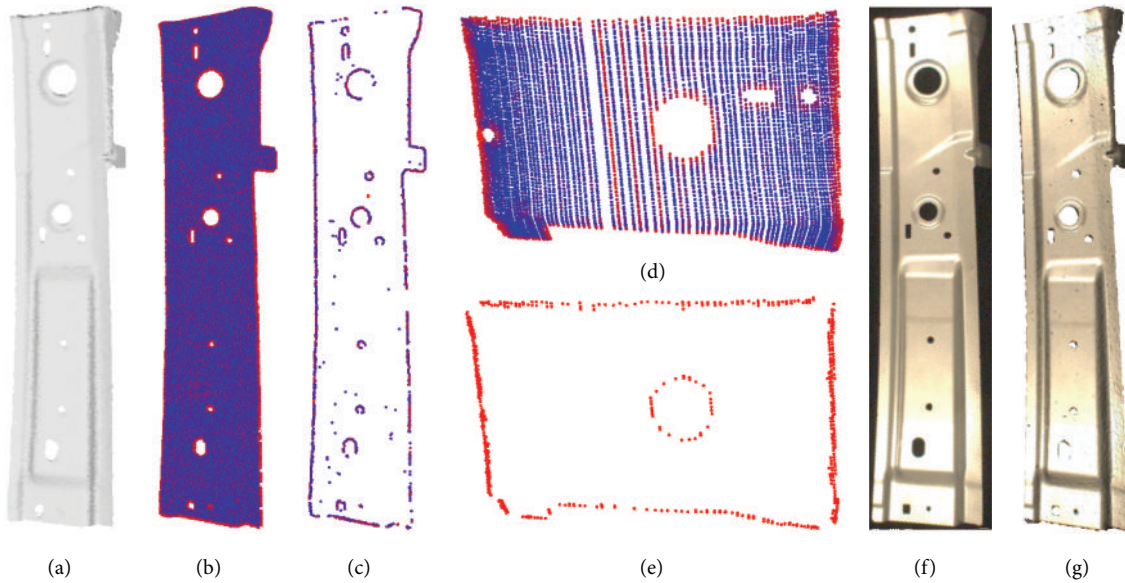


FIGURE 13: Modeling a steel sheet. ((a), (b), (c)) Reference triangle mesh, feature points, and reduced feature points (from left to right). ((d) and (e)) Partial scans of the object—feature points (d) and reduced feature points (e). ((f) and (g)) Photo/texture of the object and remeshed and texture-mapped (from left to right).

model after registration and remeshing on (e), with mapped textures on (f).

**Registration of Steel Sheets.** In this case, the problem is to map texture to a high quality 3D surface model. In order to overcome the sensor's problems with reflecting materials the object is sprayed with developer, losing the original texture information but gaining a high quality surface model. Afterwards a low quality surface model with correct texture (gained from the unsprayed object) is registered to the first model. Figure 13 shows (from (a) to (g)) the complete reference surface model, the feature points calculated from the point cloud (EVQ23, Section 3.2), and the reduced feature points (rightmost bins have been removed). In the middle the feature points before (d) and after (e) reduction of the second measurement are depicted. The texture information

(belonging to the second measurement) from a monocular shot and the result, that is, the reference model with mapped texture, are shown in Figures 13(f) and 13(g).

**7.2.2. CAD Model Registration with ToF Camera Images.** Operators in telepresence systems (as in [54]) profit from semiautonomous functions, like grasping tools for manipulation. Here, we depict a use case where our method is employed to help the humanoid robot “Justin” grasp a power screwdriver, as shown in Figure 1. The pose estimation of the screwdriver is based on one frame of a SwissRanger SR4000 ToF cam fixed on one side of the torso. As the mounting is known, the screwdriver's position on a table is assumed to be known up to 10 cm. Further, two rotational degrees of freedom (DOF) are known up to a tolerance of 40° and 120°,



respectively. In this setup the major challenge is the high noise of the ToF cam (Figure 4), which additionally produces an inaccurate extrinsic calibration.

The template feature points for global registration are calculated from CAD data. As the features we want to extract define a gap at the handle, concave regions have to be extracted. Therefore, points on the surface polygons of the CAD data are sampled and remeshed initially. On the resulting homogeneous triangle mesh, the MaNC(3) is calculated and the leftmost bins of the feature histogram are removed according to Section 3.4, until 30% of the feature points remain. Figure 5 shows the template point cloud model and the calculated feature points.

From the incoming data, that is, one frame of the ToF cam, the feature points are extracted as outlined in Section 3.5. Prior to that, the following strategy is used to handle the camera data. Considering the roughly known pose, the acquired depth image of the whole scene and its corresponding intensity image are 3D-cropped to the surrounding of the table first. Then, the table top is removed from the depth image with the help of a plane detection. Finally, the edges described in Section 3.5 are extracted to be used as features; see Figure 6.

As feature values do not correspond between template and measured data, only one feature class comprising all points is used. This works well as long as two conditions are met. First, visual and geometric features have to coincide, which is often the case with technical products. Second, in order to keep the number of false matches low, there should be not too many different edges.

Due to the possible incorporation of prior knowledge into MCR, the initial sample consists of 549 rotations on a grid, where the two rotational DOFs are sampled in steps of  $2^\circ$  and  $5^\circ$ , respectively. The pose estimation with MCR is fine adjusted with a subsequent ICP. The robot fulfills its task fluently, as the overall computing time is between 1 and 5 seconds. An example of a successfully fitted template in the original depth image is shown in Figure 1.

In order to prove the methods competitiveness it is tested against alternative methods as in Section 7.1. Here, 1000 random poses are tested using MCR and the methods from PCL. Obtaining 1000 real scans with ground truth pose is problematic. Therefore, the model is aligned to a scan manually. In order to get 1000 different poses, 1000 random rotations are applied to it. Figure 14 shows violin plots of the resulting rotational errors. Summarizing, MCR outperforms the other methods clearly. Note that the seemingly good results of SAC-IA are only due to rejected estimations outside the prior knowledge region. Only in 1% of the cases an acceptable estimation is obtained. The mean computation times for SAC-IA, SAC-IA\_full, and MCR are 0.47 s, 3.54 s, and 0.14 s, respectively, with under 0.01 s added for MCR\_ICP (if ICP correctly converged).

## 8. Experiments with SMCR

In this section, the main experiments with SMCR are carried out on an industrial robot and the method is applied to

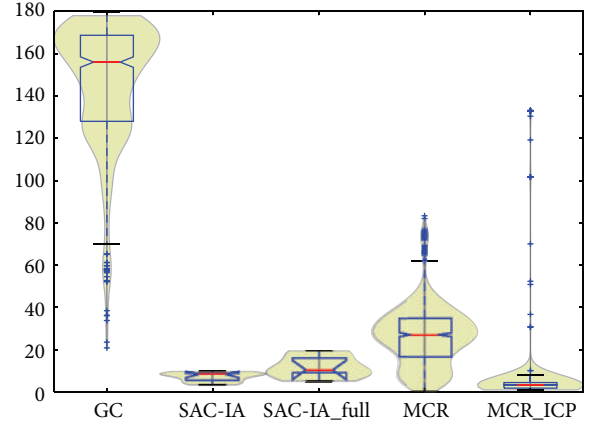


FIGURE 14: Violin plots of the rotational errors for the screwdriver compared to methods from PCL. Note that the Hough voting method never finds a solution, GC only in around 66% of the cases, and SAC-IA in around 1% of the cases (or less, when all points are used). MCR gives a result in all 1000 runs.

autonomous object modeling. In the following, the utilized hardware and experimental setup are described. Then, the results for SMCR are discussed, followed by the integration of SMCR with autonomous object modeling and an evaluation of its performance. Further, the application in mobile robot localization is depicted.

**8.1. System Setup.** Here, the utilized hardware, test objects, evaluation criteria, and parameters are described.

**8.1.1. Hardware.** For the experiments a 6 DOF industrial robot, the KUKA KR16-2, with mounted laser striper is utilized (see Figure 15). For the KR16-2, the absolute positioning error is in millimeter range. The streaming Monte Carlo registration and the autonomous object modeling are run on an external computer with Quad Xeon W3520 2.67 GHz CPUs and 6 GB RAM as the KUKA Robot Control 4 (KRC4) is not designed for additional modules. The communication between KRC4 and the external PC is performed at 250 Hz using the KUKA Robot-Sensor Interface. The laser striper is a Micro-Epsilon ScanControl 2700-100 which obtains a stripe of 640 depth points in a range of 0.3 m to 0.6 m at 50 Hz with a maximum measuring error of approx. 0.5 mm.

**8.1.2. Test Objects and Data.** All experiments in Sections 8.1–8.3 are performed for three objects: a Zeus bust, a bunny, and a wooden chevron (see Figure 16(a)). These represent different application domains, namely, cultural heritage, household, and manufacturing. The approximate height of the Zeus bust is 22 cm and that of the bunny and chevron 18 cm. Figure 16(b) shows the calculated features for these objects and Figure 16(c) the autonomously acquired 3d models (see Section 8.3).

**8.1.3. Evaluation Criteria and Parameters.** Similar to Section 7, the evaluations are done with respect to the median of rotational and translational error, denoted by  $m_t$  and  $m_R$ ,

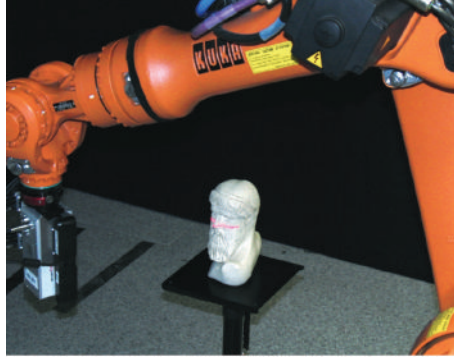


FIGURE 15: An industrial robot with attached laser striper performs a scan of an object placed onto a pedestal. The features are calculated in a real-time stream and are used to estimate the object's pose.

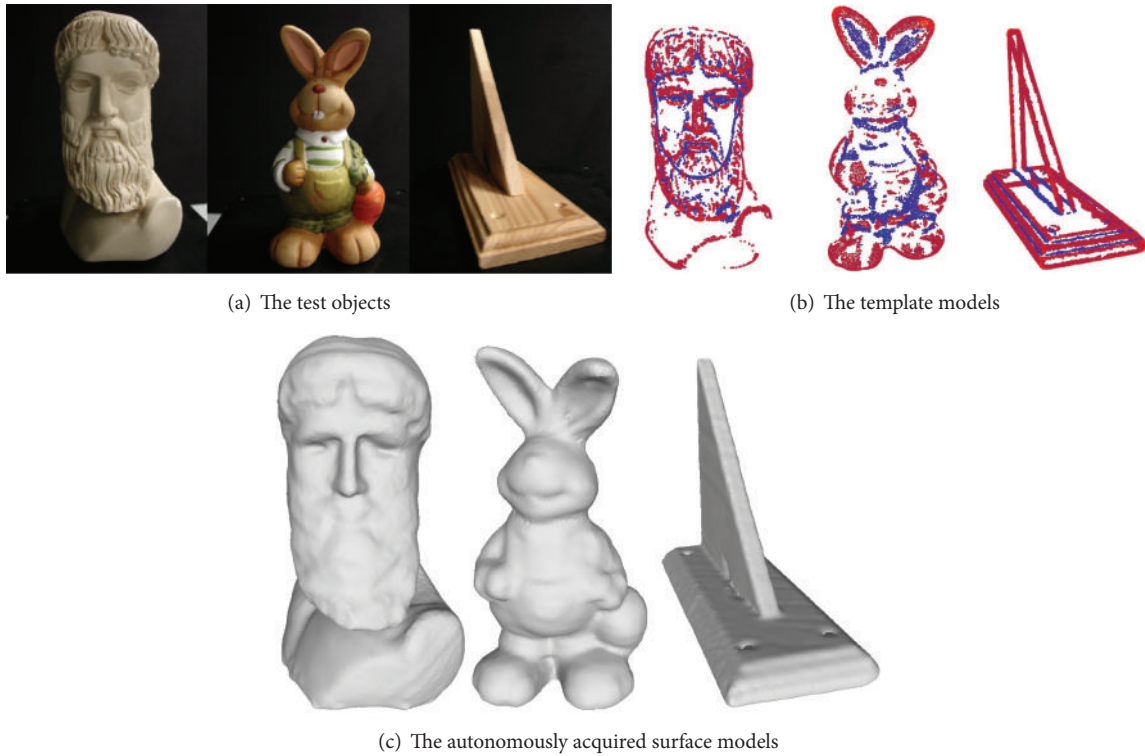


FIGURE 16: Zeus bust, bunny, and wooden chevron object (from left to right): the test objects (a) with corresponding template models (b) used for the experiments and surface models (c) acquired during autonomous object modeling (see Section 8.3.5). Colors in the template model describe the different feature classes: light/dark red occur in convex regions, blue/purple in concave regions (plane regions are removed).

respectively, and the success rate  $sr$ . Here, a success is defined, if the final error in translation and rotation is below 8 mm and  $8^\circ$ , which are tighter bounds than in Section 7.

As stated in Section 4, some parameters are adapted (by a factor of 0.8) in the iterative process. If not stated otherwise, we use a maximum number of 200 and a minimum number of 20 particles. The maximum scoring radius  $r_{\max}$  starts with 40 mm and is bounded from below by 4 mm. Neighborhood sampling of translations starts with a radius of 10 mm and is bounded by 1 mm.

**8.2. Pose Estimation with SMCR.** In this section, first an overview of the different modules of SMCR and its integration are given. Then, the influence of prior knowledge is investigated. Finally, the concept of SMCR is verified by comparison to offline global methods. Details on the results can also be found in [2].

**8.2.1. Overview.** The proposed SMCR method integrates three modules: the *3D Data Acquisition*, the *Depth Image*



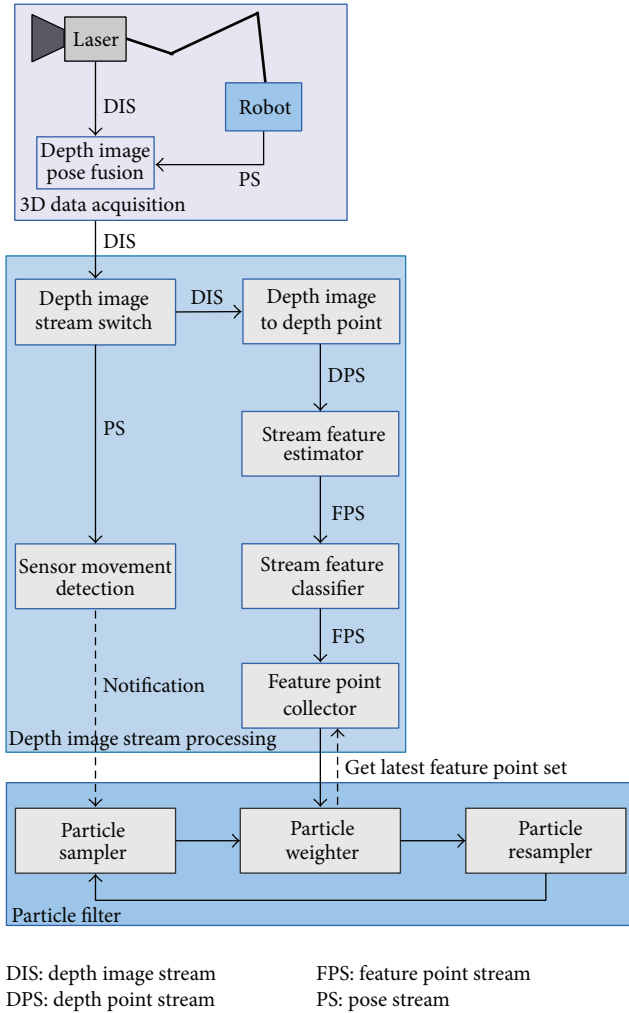


FIGURE 17: The SMCR workflow is divided into three main modules: the 3D Data Acquisition, the Depth Image Stream Processing, and the Particle Filter. Each module contains different components.

Stream Processing, and the Particle Filter module, as depicted in Figure 17.

The 3D Data Acquisition module synchronizes the depth images from the laser scanner with the pose information from the robot [55]. Therefore, the resulting depth images contain the sensor pose in robot coordinates, so that they need not be aligned to each other.

The depth image stream is handled to the Depth Image Stream Processing module. Here, the Depth Image Stream Switch component is dividing the stream again into a pose stream and a depth image stream. The pose stream is ending in the Sensor Movement Detection component, where the pose information is evaluated. If the translational or rotational difference to the last pose used for an update exceeds some predefined thresholds  $th_t$  and  $th_r$ , respectively, the sensor has moved significantly, and the particle filter is triggered to perform an update.

The depth image stream serves for feature point calculation. The depth images are converted to depth points and passed to the Stream Feature Estimator component (see

Section 3). The resulting feature point stream is handled to the Stream Feature Classifier component, which classifies the features according to the class borders of the template. Finally, the feature point stream is collected in the Feature Point Collector component from which the Particle Weighter component acquires the latest feature points on demand.

The Particle Filter module itself contains the Particle Sampler, the Particle Weighter, and the Particle Resampler components and works as described in Section 4. The Particle Sampler component starts sampling the particles when it is notified by the Sensor Movement Detection component. It performs a neighborhood sampling of the transformation particles (see Section 5). When finished, the Particle Weighter component acquires the latest feature points from the Feature Point Collector component. The particle weighting is carried out according to Section 6. After weighting, the particles are resampled with an importance resampling step by the Particle Resampler.

**8.2.2. Data and Parameters.** The method is evaluated with 10 different scan paths of the Zeus bust, 8 scan paths of the bunny, and 5 scan paths of the chevron. The difference in scan path numbers is due to the different object shapes and the chevron's symmetry. In order to ensure independent results, the scan paths are placed all around the objects. Each test is repeated 100 times to achieve a meaningful number of test runs. The tests are performed once on the real robot and the scan data is saved, as it is not feasible to repeat the whole scanning process so often. Then, the repeated tests are performed on the saved data.

The scans are registered to surface models, which are generated with a commercial 3D modeling system in advance. Feature points calculated from these models are depicted in Figure 16(b). Each feature point set is classified with 5 classes and the middle class is removed. The reduced feature point sets serve as template models and are used for registration during the laser scans. The template models of the bunny, the chevron, and the Zeus bust consist of 6714, 13075, and 8771 feature points, respectively.

The robot's pose error during a scan is usually negligible, concerning the quality of the acquired 3D models. Nevertheless, between two different scans, significant differences of robot configurations lead to considerable pose errors between the acquired 3D scan data. In between different scans, there typically occur gaps of up to 3 mm. Therefore, for each scan, a ground truth estimation is necessary because the resulting pose estimation accuracy is in the range of millimeters.

The ground truth estimations are calculated by utilizing MCR, followed by an ICP working on all acquired raw points. Correct results are assured by a visual inspection by a human operator. Moreover, the coordinate root mean square error after the ICP is checked to be lower than 0.2 mm.

**8.2.3. Influence of Prior Knowledge.** The prior knowledge about the searched transformation is separated into a translational and a rotational part. The translational part is expressed in terms of a cuboid. The volume of that cuboid is denoted by  $V(t)$  in this section. The rotational part is explained by a

TABLE 3:  $sr$ ,  $m_t$ , and  $m_R$  for different initial a priori knowledge. Fifth row: only rotations about the  $z$ -axis are sampled.

$V(t)$	$r_p$	$sr$	$m_t/m_R$
0.48 L	$10^\circ$	0.69	3.8/4.6
0.48 L	$20^\circ$	0.71	3.9/4.0
0.48 L	$45^\circ$	0.68	4.0/4.2
0.48 L	$90^\circ$	0.74	3.2/4.0
0.48 L	$90^\circ (z)$	0.93	0.7/1.0
1.22 L	$10^\circ$	0.69	4.1/4.4
1.22 L	$20^\circ$	0.64	4.3/5.9
1.22 L	$45^\circ$	0.63	4.0/5.5
1.22 L	$90^\circ$	0.71	3.5/4.1
4.00 L	$45^\circ$	0.48	7.2/8.1

mean rotation and a maximum rotational deviation from it. Additionally, the rotation axis can be assumed to be fixed in some cases, for example, if the object has been turned around the  $z$ -axis. The maximal difference from the mean rotation is denoted  $r_p$ . If not stated otherwise, the initial translations are sampled in a cuboid with an extension of  $16 \text{ cm} \times 10 \text{ cm} \times 3 \text{ cm}$ , which corresponds to the approximate position on the scanning pedestal.

In order to examine the influence of the prior knowledge, one scan of the bunny is registered to a ground truth surface model, which was acquired with a commercial scanning system. Table 3 shows the results. The initial sampling radius for the rotation seems to have little effect on the success rate and on  $m_t/m_R$ : for both fixed  $V(t)$  of 0.48 L and 1.22 L neither the success rate nor the errors are clearly increasing or decreasing with increasing radius. The volume of the cuboid for the initial translation has a clear influence if increased, as the last row shows. Both the success rate and the errors are significantly worse for a translational volume of 4.00 L compared to that of 0.48 L and 1.22 L. This confirms the suitability in autonomous 3D modeling because there is often good a priori knowledge about the position of the object, whereas the rotation cannot be told beforehand. In our scenario the rotation axis is known approximately, and often the rotation angle is restricted within a known range.

**8.2.4. Comparisons with Offline Methods.** This section shows the comparison of SMCR to MCR and SAC-IA. We chose SAC-IA as reference because it performs best in the previous experiments in Section 7. In these experiments high quality scans are used. Therefore, SAC-IA yields the best results when applied to the complete point cloud (downsampled to a density of 3 mm) with the FPFH feature, a multidimensional feature [22], which has been used in these experiments. As SAC-IA is not taking any prior information into account, it needs to perform many trials, resulting in long computation times. Introducing prior information is possible, but the method slows down a lot by this, as discussed in Section 7.2.2. Table 4 shows the result for 100 runs with all scans. Again, SAC-IA works well with 3000 iterations (results can be improved by using even more trials, but we do not find that feasible).  $Z_i, B_i, C_i$  denote the scans of the Zeus bust, the

bunny, and the chevron, respectively. The highest success rate and lowest translational and rotational error are highlighted for each scan. Overall, MCR resulted in the lowest pose errors. However, MCR took up to 30 seconds and SAC-IA up to a few minutes in the worst case whereas SMCR did not require any additional computation time.

Figure 18 highlights one of the cases where both SMCR and MCR perform poorly but SAC-IA performs relatively well, in a violin plot in (a). The  $Z_6$  scan captures a smooth surface at the back of the statue's head (on Figure 18(b)), which contains relatively few feature points. This results in larger errors of MCR and SMCR because they rely on local features. Contrary, SAC-IA uses all the points and FPFH and manages to find a good transformation in 44% of the cases. However, the distribution of the errors shows that while SAC-IA performs better, it fails completely in many cases, which influences the median not too much. In order to increase the performance of MCR and SMCR in such cases, more points could be considered. Though, more points lead to an increased computation time and introduces more uncertainty (due to a higher number of false matches).

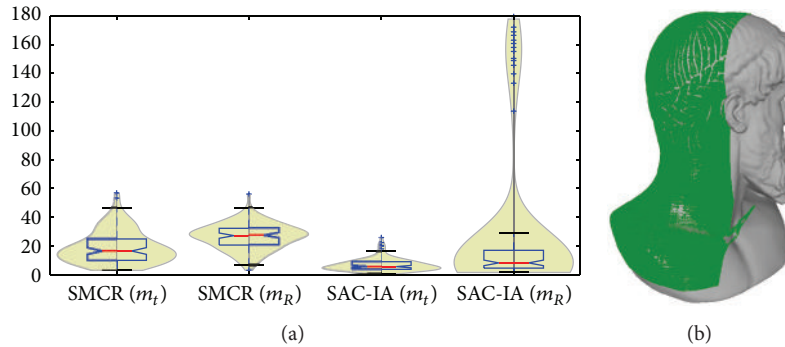
**8.3. SMCR in Autonomous Object Modeling.** In this section, first an overview is given on how the SMCR workflow (see Figure 17) is integrated into autonomous object modeling. Then, the convergence behavior and the performance of the different streaming registration variants and ICP are compared. Finally, the accuracy of autonomous modeling with integrated SMCR and repositioning is compared with the previous method.

Similar investigations can be found in [2, 3]. In contrast to those, here the particle optimization and the convergence behavior and the convergence criterion are investigated in detail. Basically, only the investigations of the uniform and Gaussian/Bingham sampling are covered by the previous publications. But for the sake of comparability with our new methods, we recomputed these tests as well in this work.

**8.3.1. Overview.** In this section, the integration of SMCR into the autonomous 3D modeling approach (see [4]) is presented. As the SMCR workflow has been described in detail in Section 8.2.1, here we concentrate on the modeling part as can be seen in the overview in Figure 19. Initially, an arbitrary laser scan of the unknown object is performed with the robot-sensor system (see Figure 15). The corresponding depth image stream contains the robot's pose information and is handled to three modules: *Mesh Update*, *Probabilistic Space Update*, and *Feature Calculation*. The features are calculated in real-time but will be used later for registration after the object has been repositioned. After updating the mesh and the probabilistic space, it is checked if the triangle mesh (surface model) has reached the desired quality, that is, if there are no holes except for not scannable parts. Then, *Next-Best-Scan Planning*, *collision-free Motion Planning*, and further laser scans are performed repeatedly, until the quality is reached. The mesh enables planning possible scan path candidates and selecting a Next-Best-Scan, in order to reach the desired surface model quality. The probabilistic space is a volumetric

TABLE 4: 100 runs:  $sr$ ,  $m_t$ , and  $m_R$  and the mean computation time  $\bar{t}$  for SMCR, MCR, and SAC-IA.

Data	SMCR			MCR			SAC-IA		
	sr	$m_t/m_R$	sr	$m_t/m_R$	$\bar{t}$	sr	$m_t/m_R$	$\bar{t}$	
$Z_1$	0.51	<b>5.7/5.8</b>	<b>0.60</b>	5.9/6.0	8.6	0.04	11.3/15.6	120.4	
$Z_2$	0.59	6.0/5.7	<b>0.92</b>	<b>3.8/3.3</b>	14.1	0.26	7.9/11.1	113.1	
$Z_3$	0.49	7.1/7.8	<b>0.58</b>	<b>4.5/6.9</b>	14.4	0.24	9.1/9.2	109.0	
$Z_4$	0.50	7.0/7.0	<b>0.80</b>	<b>2.7/4.2</b>	14.1	0.47	5.5/7.8	131.7	
$Z_5$	0.25	9.4/13.5	0.22	17.8/52.4	10.4	<b>0.92</b>	<b>2.5/3.4</b>	125.8	
$Z_6$	0.00	16.5/27.1	0.00	19.2/176.9	7.5	<b>0.44</b>	<b>5.6/8.2</b>	114.1	
$Z_7$	0.34	9.3/9.5	0.00	20.6/169.5	7.6	<b>0.58</b>	<b>5.2/6.0</b>	114.8	
$Z_8$	0.00	28.3/22.0	<b>0.82</b>	<b>6.2/3.8</b>	8.9	0.11	9.3/12.8	143.2	
$Z_9$	0.41	7.6/7.6	<b>0.69</b>	<b>4.1/4.8</b>	10.3	0.05	21.5/24.8	80.4	
$Z_{10}$	0.04	27.3/ <b>11.8</b>	0.00	23.7/124.3	6.9	<b>0.06</b>	<b>14.3/143.3</b>	79.0	
$B_1$	0.77	3.7/4.4	<b>0.90</b>	<b>2.9/2.3</b>	6.1	0.02	9.8/18.0	28.4	
$B_2$	<b>0.71</b>	3.9/4.0	0.57	<b>2.2/7.4</b>	7.4	0.06	8.2/13.0	30.3	
$B_3$	0.61	4.6/5.8	<b>0.93</b>	<b>2.4/2.5</b>	6.8	0.42	5.4/8.4	28.5	
$B_4$	0.44	3.9/9.2	<b>0.78</b>	<b>2.7/6.1</b>	4.3	0.22	6.8/11.9	23.0	
$B_5$	0.26	6.7/10.5	0.22	5.3/9.0	4.1	<b>0.70</b>	<b>4.1/5.8</b>	28.8	
$B_6$	0.15	13.9/10.1	<b>0.58</b>	<b>4.6/4.9</b>	4.4	0.20	7.5/10.4	27.7	
$B_7$	<b>0.69</b>	<b>3.8/5.6</b>	0.37	4.2/11.2	4.4	0.17	7.5/12.3	27.6	
$B_8$	0.43	6.0/ <b>8.5</b>	<b>0.45</b>	<b>4.6/8.7</b>	4.7	0.02	11.0/28.0	23.4	
$C_1$	0.15	14.8/10.5	<b>0.66</b>	<b>3.2/4.1</b>	30.8	0.15	13.9/8.1	82.5	
$C_2$	0.33	11.8/7.6	<b>0.96</b>	<b>1.4/1.7</b>	23.7	0.14	13.3/9.7	77.0	
$C_3$	0.22	14.7/6.6	<b>0.99</b>	<b>1.6/3.9</b>	16.4	0.04	19.2/13.2	63.5	
$C_4$	0.17	18.7/8.6	<b>0.52</b>	<b>7.5/4.3</b>	15.9	0.01	39.7/170.0	59.2	
$C_5$	0.06	21.8/11.0	<b>0.32</b>	<b>11.8/6.0</b>	15.1	0.00	43.0/26.8	54.5	
Units		mm/deg		mm/deg	s		mm/deg	s	

FIGURE 18: (a) Violin plots of exemplary translational errors in mm and rotational errors in degree for SMCR (left) and SAC-IA (right) for scan  $z_6$ . (b) Scan  $z_6$  (green) on ground truth model of Zeus.

model considering sensor uncertainties and giving a probability of occupancy for each voxel. It is used for exploration by *Next-Best Scan Planning* and collision avoidance during *Motion Planning*. For more details regarding the autonomous modeling and its modules we refer to [4].

As soon as the desired mesh quality has been reached, the object is repositioned in order to model previously occluded object parts. This is done by rotating it onto one of its sides. Then, a laser scan is performed along the region of interest. Again, the *Feature Calculation* is carried out on-the-fly. Synchronously, the *Particle Filter* component iteratively

performs a neighborhood sampling and weighting of the particles with the incoming feature points. After the laser scan has finished the pose estimation is instantly available. Finally, an ICP is used for fine registration, which results in a precise transformation between the original object position and the object position after repositioning. Then, the autonomous modeling is continued until a complete model is generated. In order to be able to model the object within the same coordinate system, all further generated laser scans are transformed by the resulting transformation from the registration.

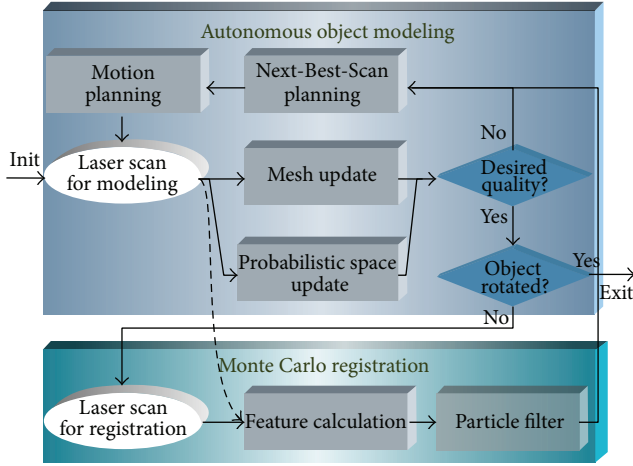


FIGURE 19: Overview of the integration of autonomous object modeling with streaming Monte Carlo registration. Gray boxes represent modules, white ovals robot-sensor system actions, and blue diamond boxes decisions.

### 8.3.2. SMCR and ICP with Partially Overlapping Submodels.

In order to show that local methods like ICP cannot be applied, we register two partially overlapping scans of the Zeus bust to each other, which poses a higher challenge than registering a submodel to a complete object model with low noise. For this purpose, we record 20 different pairs of overlapping scans (see Figure 20 for an example). To one of the scans in each pair we apply 10 different random transformations with translational and rotational variations. The translation vectors have a maximal norm of 20 mm and the rotations a maximum rotation angle of  $45^\circ$ ,  $90^\circ$ ,  $120^\circ$ , or  $180^\circ$  around the  $z$ -axis. The prior knowledge in SMCR is set accordingly, denoted with  $Z45X$ ,  $Z90X$ ,  $Z120X$ ,  $Z180X$ , respectively in Table 5.  $Z$  stands for the object Zeus and  $X$  presents the different methods described below. After the SMCR, we apply an ICP with a small search radius (20 mm) and few iterations for fine fitting, denoted by SMCR-ICP. We compare the results to a pure ICP with a bigger search radius (50 mm), simply denoted by ICP. For each rotation range we test the original Uniform neighborhood sampling  $U$ , the proposed Gaussian sampling  $G$ , and the Gaussian sampling with Optimization step  $O$ , denoted by a capital  $U$ ,  $G$ , or  $O_i$  in the data names, respectively. For instance,  $Z90N$  denotes the case of uniformly sampled rotations with a maximal rotation angle of  $90^\circ$  for the Zeus bust, and  $B120O_5$  denotes the case of optimization in every 5th step (normal/Bingham sampling) and a maximal rotation angle of  $120^\circ$  for the bunny. In the table, the highest success rate and lowest translational and rotational error are highlighted for each rotation angle over all methods. Overall, the accuracy in translation and the success rate are increased for SMCR when applying the optimization step. The rotational accuracy is not generally increased. The effects appear in all rotation ranges. Concerning pure ICP, it becomes clear that, with increasing rotation range, its performance significantly decreases, both in accuracy and reliability. SMCR-ICP outperforms pure ICP clearly, both in reliability and accuracy. Further,

the performance of pure SMCR is also significantly better than ICP. The frequency of the optimization steps has no clear effect on the results, neither on pure SMCR nor on the combination with ICP. Uniform sampling yields slightly better results than Bingham/Gaussian sampling.

Success rates of about 0.6 of pure SMCR appear to be pretty small for two reasons: on the one hand, the parameters are not tuned for the data sets. On the other hand, the partially overlapping scans are harder to register as it seems at first glance. The most descriptive features are not easy to scan and very similar features are spread over the object. Moreover, the descriptive features appear more or less randomly in the data sets because the scan paths are chosen arbitrarily. The results in the preceding sections show that with this kind of data other state-of-the-art methods perform even worse, even when registering to a complete high-precision ground truth surface model.

### 8.3.3. Convergence Behavior for SPFR, SMCR, and SMCRO.

In this section, the convergence of SMCRO, SMCR, and SPFR is investigated. In order to account for the application in autonomous 3D modeling, we first autonomously acquire a more or less complete model of the bunny and the Zeus bust (without bottom part). Then, we reposition them on the scanning pedestal and acquire one scan manually. With this scan, 1000 repetitions for pose estimation are performed.

The ground truth estimation is calculated by utilizing MCR, followed by an ICP working on all acquired raw points of the ten subscans. Correctness is assured by visual inspection of a human operator.

Therefore, we repeat estimations for one scan path of the bunny and the Zeus bust 1000 times and calculate the medians of the translational and rotational errors in each update step, as depicted in Figure 21. In the case of SMCRO, the optimization is carried out in every step and every 2nd, every 5th, or every 10th step. Clearly, SMCR yields better convergence behavior than SPFR, which does not reach the success criterion at all. SMCRO in turn converges faster than SMCR, and the faster the more optimization steps are performed. Note that, at the update steps, the error medians often visibly drop down.

However, the optimization needs to be carried out with caution as, in individual cases, for too early or too many optimization steps the method may tend to converge to the wrong transformation, especially for objects with many symmetries. Therefore we started optimization not before the 5th step in any case.

### 8.3.4. SMCR and ICP in Autonomous Modeling.

In this section, an extensive evaluation of SPFR, SMCR, and SMCRO and comparison to ICP are performed. Therefore, a more or less complete model (without bottom part) of the object is autonomously acquired initially. Then, the object is repositioned on the scanning pedestal and 10 different single scans are performed manually. Each of the manual scans is transformed by ten different random transformations, giving a total of 100 different test scans for each object. These are registered to the corresponding previously acquired complete



TABLE 5:  $m_t$ ,  $m_R$ , and  $\bar{t}$  for SMCR with Gaussian (G)/uniform (U) sampling and with optimization (O) step and ICP for 200 tests of partially overlapping scans of the Zeus bust and 45°, 90°, 120°, and 180° rotations.

Data	SMCR		SMCR-ICP			ICP		
	$m_t/m_R$	sr	$m_t/m_R$	sr	$\bar{t}$	$m_t/m_R$	sr	$\bar{t}$
Z45G	6.3/5.0	0.5	1.7/2.5	0.7	2.1			
Z45U	6.3/6.3	0.5	1.5/2.7	0.7	2.0			
Z45O <sub>1</sub>	2.7/3.4	0.6	1.5/2.3	<b>0.8</b>	1.6	2.8/6.6	0.5	7.1
Z45O <sub>2</sub>	2.9/3.1	0.6	1.6/2.4	0.7	1.8			
Z45O <sub>5</sub>	2.7/3.2	0.6	<b>1.4/2.4</b>	<b>0.8</b>	1.7			
Z45O <sub>10</sub>	2.5/2.9	0.6	<b>1.4/2.3</b>	<b>0.8</b>	1.8			
Z90G	7.1/7.8	0.4	1.5/2.7	0.7	2.0			
Z90U	5.7/7.1	0.5	1.7/3.0	0.7	2.3			
Z90O <sub>1</sub>	3.1/3.8	0.6	1.7/2.5	0.7	1.7	12.6/41.9	0.3	7.2
Z90O <sub>2</sub>	3.2/4.7	0.5	1.6/2.4	0.7	2.0			
Z90O <sub>5</sub>	3.7/4.2	0.6	1.5/2.4	0.7	1.8			
Z90O <sub>10</sub>	3.8/4.5	0.5	<b>1.4/2.3</b>	<b>0.8</b>	1.9			
Z120G	6.0/5.3	0.5	1.6/2.8	0.7	2.2			
Z120U	5.7/6.9	0.5	1.6/2.7	0.7	1.9			
Z120O <sub>1</sub>	3.7/3.7	0.6	1.5/2.7	0.7	1.8	12.6/53.5	0.2	6.9
Z120O <sub>2</sub>	2.6/4.8	0.6	<b>1.3/2.5</b>	<b>0.8</b>	1.7			
Z120O <sub>5</sub>	2.7/2.7	0.6	<b>1.3/2.3</b>	<b>0.8</b>	1.6			
Z120O <sub>10</sub>	2.7/4.0	0.6	1.5/2.6	0.7	1.8			
Z180G	5.3/5.6	0.5	1.6/2.6	0.7	2.1			
Z180U	4.1/4.9	0.6	1.6/2.3	<b>0.8</b>	1.6			
Z180O <sub>1</sub>	2.9/4.0	0.6	1.7/2.2	0.7	1.8	15.2/71.6	0.2	3.7
Z180O <sub>2</sub>	4.0/5.6	0.5	1.6/2.5	0.7	1.9			
Z180O <sub>5</sub>	2.7/3.0	0.6	1.5/2.4	0.7	1.6			
Z180O <sub>10</sub>	1.8/3.4	0.6	<b>1.4/2.4</b>	0.7	1.7			
Units	mm/deg		mm/deg		s	mm/deg		s

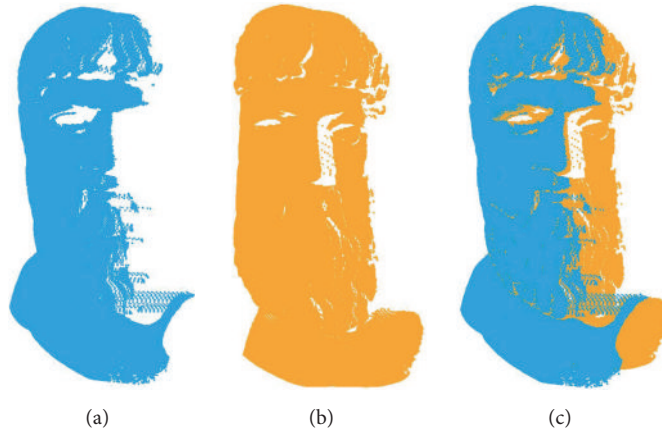


FIGURE 20: Two typical scans of Zeus bust before ((a) and (b)) and after alignment (c).

model. The ground truth estimation is calculated by MCR and an ICP working on the raw points of the ten subscans. Visual inspection of a human operator assures the correctness.

*Goodness of Fit and Success Rates.* Table 6 shows the results of the experiments concerning rotational and translational error as well as the success rate. In contrast to the experiments in

Section 8.3.2, the registration is performed based on the 3D model. Additionally we perform them for the bunny (denoted by a capital *B*) and the chevron (*C*) objects.

The results of the bunny and the Zeus bust clearly show that the proposed optimization step yields a higher accuracy and success rate. Moreover, in many cases, results get better with increasing optimization frequency. Finally, the errors



TABLE 6:  $m_t$ ,  $m_R$ , and  $\bar{t}$  for SMCR with Gaussian (G)/uniform sampling (U), with optimization step (O), with aborting after convergence (C) and ICP for the Zeus bust (Z), the Bunny (B), and the Chevron (C) and 45°, 90°, 120°, and 180° rotations.

Data	SMCR		SMCR-ICP			ICP		
	$m_t/m_R$	sr	$m_t/m_R$	sr	$\bar{t}$	$m_t/m_R$	sr	$\bar{t}$
Z45G	3.1/2.3	0.7	0.4/0.4	0.9	1.6	0.9/1.8	0.6	10.9
Z45U	2.8/1.8	0.7	<b>0.3</b> /0.4	0.9	1.4			
Z45O <sub>1</sub>	0.9/0.4	0.8	0.4/0.4	0.9	2.6			
Z45O <sub>2</sub>	1.1/0.6	0.8	0.4/ <b>0.3</b>	<b>1.0</b>	2.9			
Z45O <sub>5</sub>	1.3/1.3	0.8	0.4/0.4	<b>1.0</b>	3.4			
Z45O <sub>10</sub>	1.3/1.1	0.8	0.4/ <b>0.3</b>	0.9	3.5			
Z45C	1.3/0.9	0.9	<b>0.3</b> /0.4	<b>1.0</b>	1.3		0.6	10.6
Z90G	5.3/4.3	0.6	0.5/0.5	0.8	1.7	2.0/11.3	0.4	9.7
Z90U	2.7/1.8	0.6	0.4/0.4	<b>0.9</b>	1.7			
Z90O <sub>1</sub>	0.9/ <b>0.4</b>	0.8	<b>0.4</b> /0.4	<b>0.9</b>	2.9			
Z90O <sub>2</sub>	1.1/0.5	0.8	<b>0.4</b> /0.4	<b>0.9</b>	3.2			
Z90O <sub>5</sub>	1.5/1.4	0.8	<b>0.4</b> /0.4	0.8	3.5			
Z90O <sub>10</sub>	2.6/1.7	0.7	<b>0.4</b> /0.4	0.8	4.4			
Z90C	1.1/0.8	0.8	<b>0.4</b> /0.4	<b>0.9</b>	1.3		0.4	10.4
Z120G	5.3/4.6	0.5	0.5/0.5	0.8	1.8	7.2/41.5	0.4	10.6
Z120U	4.5/3.0	0.6	0.6/0.4	0.8	1.4			
Z120O <sub>1</sub>	0.8/ <b>0.3</b>	0.8	<b>0.4</b> /0.4	<b>0.9</b>	2.7			
Z120O <sub>2</sub>	1.2/0.6	0.8	<b>0.4</b> /0.4	<b>0.9</b>	2.5			
Z120O <sub>5</sub>	1.3/1.4	0.7	<b>0.4</b> /0.4	<b>0.9</b>	2.4			
Z120O <sub>10</sub>	2.2/1.9	0.6	<b>0.4</b> /0.4	0.8	2.7			
Z120C	1.3/1.0	0.7	<b>0.4</b> /0.4	0.8	1.4			
Z180G	8.8/6.2	0.5	0.7/0.5	<b>0.8</b>	1.8	10.8/51.9	0.3	10.3
Z180U	6.1/6.4	0.5	0.7/0.5	0.7	1.7			
Z180O <sub>1</sub>	1.0/0.5	0.7	<b>0.4</b> /0.5	<b>0.8</b>	2.4			
Z180O <sub>2</sub>	1.3/0.9	0.7	<b>0.4</b> /0.5	<b>0.8</b>	2.4			
Z180O <sub>5</sub>	1.5/2.0	0.7	<b>0.4</b> /0.5	<b>0.8</b>	2.5			
Z180O <sub>10</sub>	2.6/2.0	0.6	<b>0.4</b> /0.5	<b>0.8</b>	2.7			
Z180C	1.6/1.1	0.6	0.5/ <b>0.4</b>	<b>0.8</b>	1.5			
B45G	7.1/8.3	0.3	0.7/ <b>0.2</b>	0.9	1.0	<b>0.6</b> /0.3	<b>1.0</b>	2.6
B45U	6.7/7.0	0.6	0.7/ <b>0.2</b>	<b>1.0</b>	0.9			
B45O <sub>1</sub>	0.7/0.3	0.9	0.7/ <b>0.2</b>	<b>1.0</b>	2.0			
B45O <sub>2</sub>	0.7/0.5	0.9	0.7/ <b>0.2</b>	0.9	0.9			
B45O <sub>5</sub>	1.0/1.3	0.9	0.7/ <b>0.2</b>	0.9	0.9			
B45O <sub>10</sub>	3.1/4.4	0.7	0.7/ <b>0.2</b>	<b>1.0</b>	0.9			
B45C	0.8/0.8	0.9	0.7/ <b>0.2</b>	0.9	1.0			
B90G	7.4/8.9	0.3	<b>0.7</b> /0.2	<b>0.9</b>	1.0	<b>0.7</b> /0.3	0.8	2.9
B90U	6.7/8.5	0.4	<b>0.7</b> /0.3	<b>0.9</b>	0.9			
B90O <sub>1</sub>	<b>0.7</b> /0.4	0.9	<b>0.7</b> /0.2	<b>0.9</b>	2.3			
B90O <sub>2</sub>	<b>0.7</b> /0.8	0.8	<b>0.7</b> /0.2	<b>0.9</b>	2.6			
B90O <sub>5</sub>	1.4/2.3	0.8	<b>0.7</b> /0.2	<b>0.9</b>	2.7			
B90O <sub>10</sub>	4.2/6.5	0.6	<b>0.7</b> /0.2	<b>0.9</b>	3.0			
B90C	0.8/0.8	0.8	<b>0.7</b> /0.2	<b>0.9</b>	1.1			
B120G	8.3/10.3	0.3	<b>0.7</b> /0.2	<b>0.9</b>	1.1	0.8/0.3	0.7	3.0
B120U	6.9/7.6	0.4	<b>0.7</b> /0.2	<b>0.9</b>	1.0			
B120O <sub>1</sub>	<b>0.7</b> /0.4	0.9	<b>0.7</b> /0.2	<b>0.9</b>	2.4			
B120O <sub>2</sub>	0.8/0.7	0.8	<b>0.7</b> /0.2	<b>0.9</b>	2.4			
B120O <sub>5</sub>	1.9/2.7	0.8	<b>0.7</b> /0.2	<b>0.9</b>	2.9			
B120O <sub>10</sub>	4.2/6.1	0.6	<b>0.7</b> /0.2	<b>0.9</b>	3.3			
B120C	0.8/0.8	0.9	<b>0.7</b> /0.2	<b>0.9</b>	1.2			
B180G	9.3/11.9	0.2	<b>0.7</b> /0.2	<b>0.9</b>	1.2	12.3/128.5	0.4	3.1
B180U	8.1/11.7	0.3	<b>0.7</b> /0.2	<b>0.9</b>	1.1			
B180O <sub>1</sub>	0.7/0.4	0.8	<b>0.7</b> /0.2	<b>0.9</b>	2.5			
B180O <sub>2</sub>	0.8/0.7	0.8	<b>0.7</b> /0.2	<b>0.9</b>	2.6			
B180O <sub>5</sub>	1.6/2.2	0.7	<b>0.7</b> /0.2	0.8	2.7			

TABLE 6: Continued.

Data	SMCR		SMCR-ICP			ICP		
	$m_t/m_R$	sr	$m_t/m_R$	sr	$\bar{t}$	$m_t/m_R$	sr	$\bar{t}$
B180O <sub>10</sub>	5.4/9.0	0.4	<b>0.7/0.2</b>	<b>0.9</b>	3.4			
B180C	0.8/0.9	0.8	<b>0.7/0.3</b>	0.9	1.2			
C45G	12.3/0.9	0.4	2.2/0.9	0.6	1.9			
C45U	9.5/0.8	0.5	1.9/1.1	0.7	1.9			
C45O <sub>1</sub>	5.8/1.1	0.5	2.3/1.2	0.6	2.2	<b>1.7/0.8</b>	<b>0.9</b>	7.1
C45O <sub>2</sub>	5.1/1.3	0.5	2.2/1.2	0.6	2.3			
C45O <sub>5</sub>	8.9/1.8	0.5	2.2/1.2	0.6	2.5			
C45O <sub>10</sub>	5.8/1.5	0.6	1.8/1.1	0.8	2.6			
C45C	13.0/2.3	0.4	4.8/1.3	0.5	1.9			
C90G	11.4/1.0	0.4	2.2/1.0	0.6	1.9			
C90U	15.5/1.1	0.3	2.2/1.3	0.6	1.9			
C90O <sub>1</sub>	4.0/1.2	0.6	2.2/1.2	0.7	2.2	<b>1.7/0.9</b>	<b>0.9</b>	8.0
C90O <sub>2</sub>	4.9/1.1	0.6	2.2/1.2	0.6	2.3			
C90O <sub>5</sub>	7.0/1.9	0.5	2.6/1.3	0.6	2.5			
C90O <sub>10</sub>	9.8/1.4	0.5	2.2/1.1	0.6	2.7			
C90C	15.4/2.7	0.3	9.6/1.4	0.5	1.9			
C120G	15.3/1.1	0.3	7.4/1.3	0.5	1.7			
C120U	13.8/1.2	0.4	<b>2.2/1.3</b>	0.6	1.8			
C120O <sub>1</sub>	10.8/1.1	0.5	2.5/1.3	0.6	2.2	<b>2.4/1.0</b>	<b>0.7</b>	7.0
C120O <sub>2</sub>	6.4/1.2	0.5	<b>2.2/1.2</b>	0.6	2.2			
C120O <sub>5</sub>	11.8/1.8	0.4	3.9/1.3	0.5	2.3			
C120O <sub>10</sub>	14.2/1.4	0.4	2.6/1.0	0.5	2.5			
C120C	14.1/2.8	0.4	8.3/1.3	0.5	1.9			
C180G	18.4/5.2	0.1	12.3/2.1	0.4	1.7			
C180U	17.0/3.0	0.2	11.2/2.1	0.4	1.6			
C180O <sub>1</sub>	14.0/1.1	0.3	11.6/2.1	0.4	2.1	6.0/1.5	<b>0.6</b>	7.3
C180O <sub>2</sub>	13.5/1.5	0.4	9.5/1.3	0.5	2.3			
C180O <sub>5</sub>	16.5/2.3	0.2	12.0/2.0	0.4	2.5			
C180O <sub>10</sub>	13.6/1.9	0.3	<b>2.5/1.3</b>	<b>0.6</b>	2.6			
C180C	15.3/3.6	0.2	12.3/2.1	0.4	1.8			
Units	mm/deg		mm/deg		s	mm/deg		s

with SMCR-ICP are only slightly smaller than with the optimization in every step.

SMCR itself performs good, but accuracy is not comparable to the ICP (if both are successful). The proposed Gaussian sampling does not lead to higher accuracy or success rates. The accuracy as well as the success rate is not much influenced in the example of the bunny. The opposite is true for the Zeus bust. The pure ICP is working surprisingly good, especially with the chevron, though it gets unusable for rotation angles over 45° for the bust. SMCR-ICP works extremely good, even in cases when SMCR yields problematic results.

Concerning the chevron, the rotation is estimated pretty good by all methods and the translation very bad especially when allowing for large rotations. Note that the low median in the ICP results is misleading, as the success rates are pretty low, compared to the results with the Zeus bust and the bunny. An explanation could be that on the one hand the chevron has big flat surface areas which allow a robust estimation of rotations. On the other hand, this seems to allow the translation to slide along these areas, especially vertically along the triangular part. Additionally, there are a lot of spurious measurements, including the pedestal the object is placed on.

*Convergence Criterion.* The results obtained with the application of the convergence criterion (SMCRC) are denoted with a capital C as last letter in the table rows of Table 6; for example, Z45C denotes the experiments with the Zeus bust for an angle of 45° and with abortion due to convergence. In the tests, we used convergence thresholds of  $c_t = 2$  mm and  $c_r = 2^\circ$  (see Section 6.2.3). Concerning the Zeus bust and the bunny, Table 6 shows that the results with SMCRC are comparable with those of SMCRO<sub>2</sub>, and after application of ICP comparable to SMCRO<sub>1</sub>. Further, SMCRC clearly outperforms the variants with few optimization steps (SMCRO<sub>5</sub> and SMCRO<sub>10</sub>). Concerning the chevron, SMCRC yields clearly worse results than SMCRO. Probably this is due to the geometry of the chevron, where initially a large translational error still allows for a very good matching of wrong correspondences.

*8.3.5. Autonomous Object Modeling.* Here, we compare the autonomous modeling results without repositioning the object as in [4] with the integration of the SMCR and repositioning of the object as presented in Figure 19. Therefore, the complete object modeling is performed 10 times each for

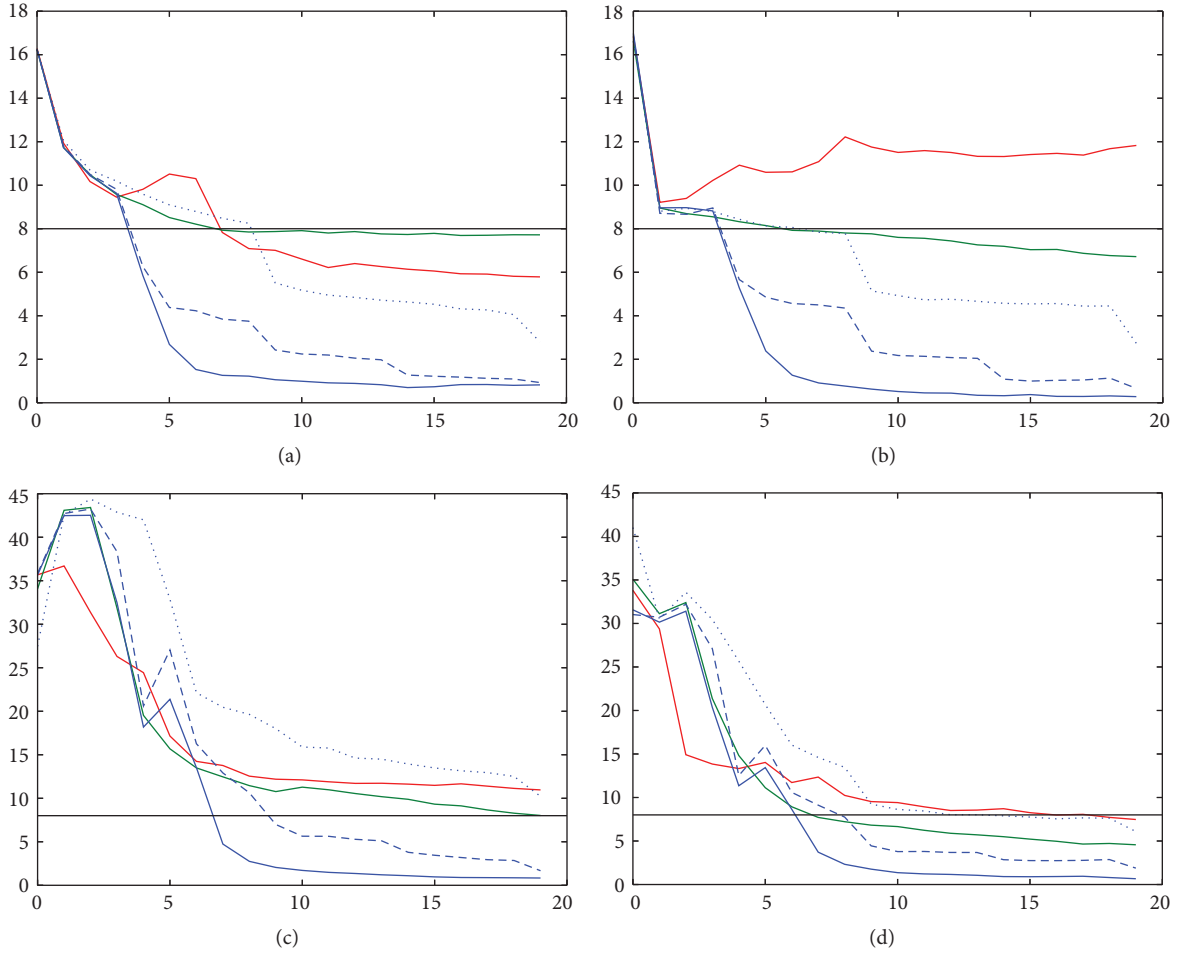


FIGURE 21: Exemplary error convergence of SPFR (red), SMCR (green), and SMCRO (blue) for 1000 runs on a bunny (top) and a Zeus (bottom) scan. Optimization is performed in every (solid) step and every 5th (dashed) and every 10th (dotted) step. Left: translational error in mm. Right: rotational error in degree. x-axis: step number. The black horizontal line represents the success threshold of 8 degrees or 8 mm.

the bunny, the Zeus bust, and the wooden chevron. For a comparison of the autonomous modeling method [4] with the other state-of-the-art methods concerning the algorithms for NBV planning, we refer to [56]. It has been shown that the NBV approach which plans NBVs based on the boundaries of the surface models and considers information gain and surface quality for the NBV selection outperforms the other methods.

During these experiments, the object is manually placed onto its side after the desired quality for the visible object parts has been reached. For the 10 runs, different arbitrary initial scans and variations in the repositioning object orientation are chosen. The average model completeness and coordinate root mean square (CRMS) error when comparing with ground truth models are given in Table 7. The completeness is evaluated by comparing a ground truth model with the generated triangle mesh. The CRMS gives a measure for the model error which is influenced by the fact that details in the object are not modeled perfectly as can be seen in Figure 16(c). The error is mainly a result of sensor noise, sensor calibration, and robot accuracy which for

the KUKA KR16-2 is in millimeter range. The completeness after repositioning is larger as the bottom parts have been filled. Figure 22 shows exemplarily for the Zeus bust how the bottom part is filled accurately with no major deviations due to the different object positions. The completeness still does not reach 100% which is due to the NBS planning which aborts based on a coverage estimation utilizing the current surface model which sometimes is noisy. However, these are just small holes which can easily be filled by a postprocessing technique. For the bunny and chevron, 100% is reached for some runs whereas for the Zeus bust a small part in the chin area below the beard could never be filled due to sensor restrictions as this area is very narrow. The CRMS shows that, due to object repositioning and SMCR, the model error does not increase. The CRMS is even slightly lower when the object is repositioned. One reason for this is probably due to the fact that along borders in the mesh larger errors occur due to incorrect matching (see Figure 22(a)). Further, the objects do not have many details on the bottom and thus the error is lower which influences the average error positively.

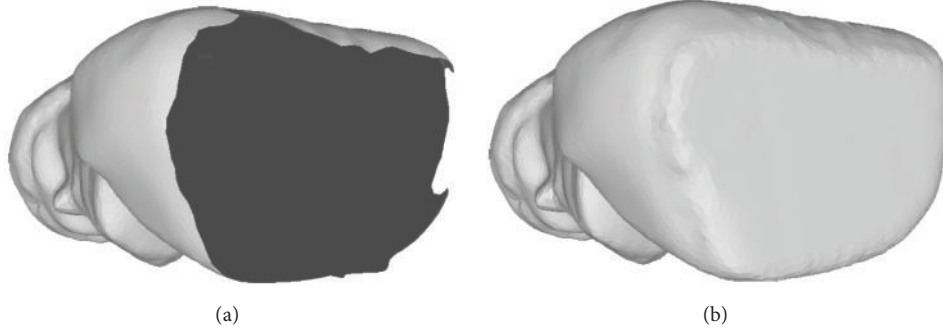


FIGURE 22: 3D model of Zeus bust from bottom view without (a) and with repositioning the object and performing SMCB (b).

TABLE 7: Comparison of modeling results without and with repositioning using SMCB (average of 10 runs).

Object	Repositioning	Completeness	CRMS
Zeus	No	88.0%	1.56 mm
	Yes	97.3%	1.46 mm
Bunny	No	91.7%	1.56 mm
	Yes	99.7%	1.37 mm
Chevron	No	97.7%	1.44 mm
	Yes	99.9%	1.34 mm



FIGURE 23: Kidnapped robot problem: a robot is randomly placed in a predefined area (green) in a known map. After self-localization the robot plans a path (yellow) to its goal.

**8.4. SMCRO in Mobile Robot Localization.** In order to compare SMCRO with a standard particle filter based on 3D depth images (see [15]), we set up a simulated kidnapped robot scenario, where a mobile robot is randomly placed in a predefined area at an unknown pose  $(x, y, \theta) \in \mathbb{R}^2 \times [-\pi, \pi]$  in a given global 3D map as shown in Figure 23. The particle filter weights the particles with a likelihood representing independently and identically normally distributed errors for

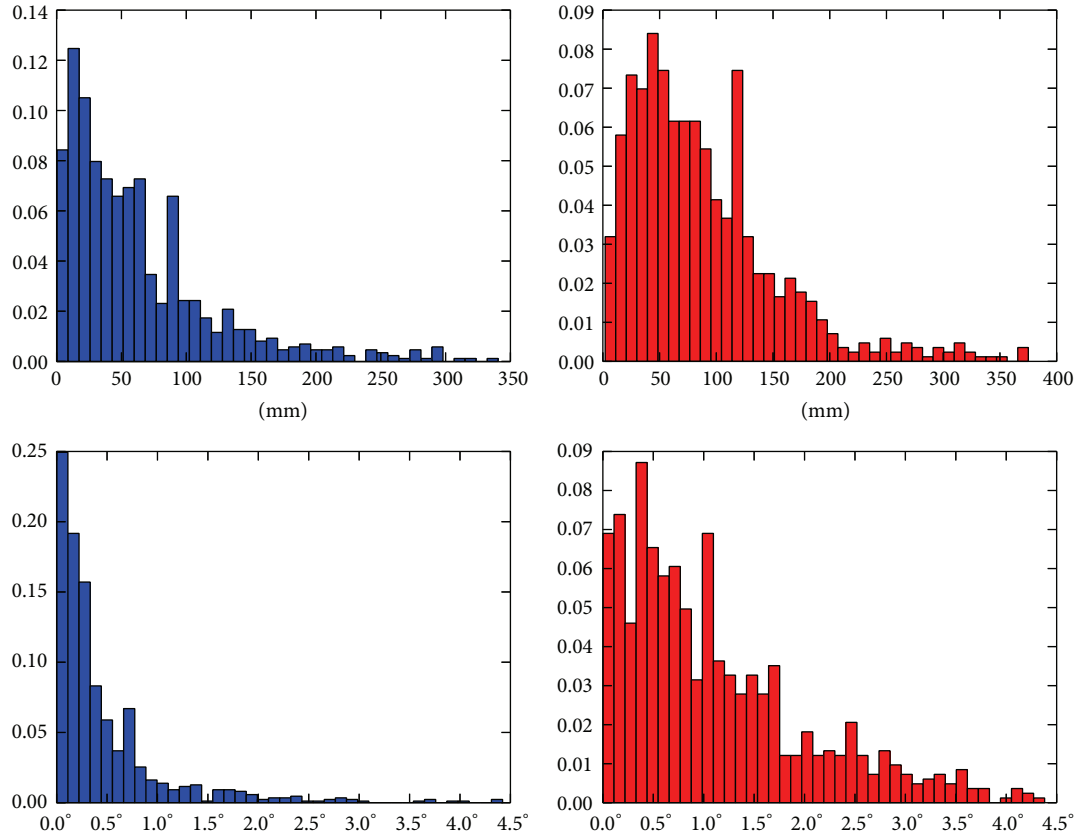
the depth values. Note that the map is 3D and the depth image simulation (for expected depth images in the weighting step) is also done in 3D, whereas the pose estimation is only in 2D. Our robot is moving omnidirectional on four wheels and is equipped with eight ToF cameras, each having a field of view of about  $35^\circ \times 45^\circ$  and a resolution of  $48 \times 64$  pixels. This setup is chosen such to represent the KUKA OmniRob, equipped with eight O3D100 ToF cameras of ifm.

The robot's task is to get "home" immediately, which can be divided into three subtasks: at first the robot actively localizes itself using the ToF cameras and a particle filter to get its current pose in the global map. After a successful localization the robot plans a piecewise linear path to reach the goal, containing about 20 waypoints. Finally the robot moves to the goal along the calculated path. For the odometry as well as the ToF data artificial noise is added. At every waypoint the robot corrects its odometrical pose using the ToF cameras to reduce the dead reckoning error. For correcting the pose, the standard particle filter is used. SMCRO is running in parallel, enabling a comparison of pose estimates, starting with the initial localization.

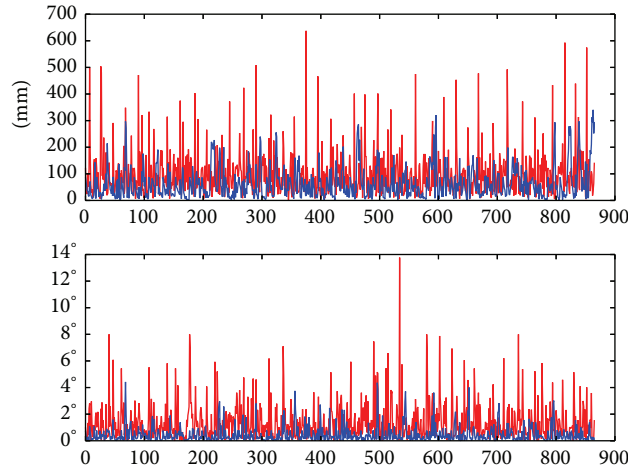
In order to achieve a better comparison the whole task has been repeated 40 times, resulting in a total number of 866 pose estimation steps. Figure 24(b) shows the errors of all these steps, starting with the first run. Clearly a periodic behavior is seen, which resembles the strong dependency of the pose estimation quality from the real pose in the environment. Note that the histograms (see Figure 24(a)) have been cut off on the right for better visibility. Both histograms and plots show that the accuracy of SMCRO outperforms that of the standard particle filter. The medians of translational and rotational errors are 76.4 mm and  $0.94^\circ$  for the standard particle filter and 47.6 mm and  $0.26^\circ$  for SMCRO. The maximum of translational and rotational errors are 637.5 mm and  $13.8^\circ$  for the standard particle filter and 340.5 mm and  $4.4^\circ$  for SMCRO.

Note that the mean computation times are 1.59 s for the standard particle filter and 2.1 s for SMCRO. In contrast to SMCRO, the standard particle filter has already been optimized for the mobile robot application. The number of beams from the depth images used for the updates are dependent on the number of particles in order to assure acceptable computation times.





(a) Histograms of errors of all iterations



(b) Errors of all update steps of all iterations, sorted by runs

FIGURE 24: Comparison between SMCRO (blue) and a standard particle (red) based on depth images. In the plots the errors are sorted by runs, resulting in periodic effects.

**8.5. Discussion.** The comparison with offline global methods yields no definite best method. But they clearly show that the method is competitive in accuracy and robustness with available state-of-the-art methods in many cases. The advantage of no extra computation time is clear, as even for these simple objects the offline methods need up to 2.5 minutes for getting similar results, whereas our method does not need any extra computation time. This effect will dramatically increase

with larger objects, which could not be investigated with the hardware setup in this paper, due to kinematic constraints.

The comparison between uniform and normal/Bingham sampling show that normal/Bingham sampling does neither improve accuracy nor robustness. The investigation of the different scoring variants (SPFR, SMCR, SMCRO, and SMCRC) shows that SMCR has a clearly better convergence behavior than SPFR. The optimization in SMCRO yields

faster convergence, higher accuracy, and higher success rate. In many cases, it is advantageous and poses no problem to optimize in every weighting step. However, this has to be done carefully, as convergence to false transformations can occur. In the data sets of this paper, no delay in updating occurred when updating in every step. However, in bigger data sets, it could lead to problems concerning computation time. The given convergence criterion proved to work efficiently when combined with optimization, with only slightly lower accuracy and robustness.

For autonomous modeling with SMCR, the results show that almost complete 3D models including object parts which are not visible in the initial pose can be created. Further, the average model error when comparing to ground truth is not increased by the object repositioning and SMCR which shows that the pose estimation is performed accurately for all runs.

First experiments in mobile robot localization show that SMCR is able to achieve a significantly higher pose accuracy than a tuned standard particle filter with only slightly higher computation time which can easily be optimized.

## 9. Conclusion and Future Work

In this work, Monte Carlo registration methods have been presented and advanced. The offline particle filter variant searching in the space of rotations outperformed the state-of-the-art algorithms, especially when prior knowledge is available. The proposed curvature features proved to be robust under sensor noise. For streaming application, the scoring of rotations is too time consuming. Thus, the space of rigid body transformations is searched in this case. Various real data experiments showed the competitiveness of the streaming variant. Thereby, convergence behavior and influence of prior knowledge have been investigated. The streaming variant has been enhanced with pose optimization and convergence criterion. The applicability in autonomous 3D modeling has been proven by various experiments with an industrial robot and a laser striper. The integration of the streaming registration into autonomous object modeling worked robustly and allowed for obtaining complete high quality 3D surface models of initially unknown objects. Finally, experiments in mobile robot localization showed that the straightforward application without any tuning yielded a higher accuracy than a standard particle filter (with comparable computation time).

Future work will focus on autonomous feedback of failure, in order to enable rescanning and detailed investigation of other convergence criteria. Furthermore, we want to apply the method during modeling of object scenes as presented in [56] where the template models contain less data as objects are occluded by others. Moreover, we want to apply the method to real mobile robots for localization and for modeling of larger indoor areas of buildings with big data sets. If data becomes too big to keep all feature points in memory, probably the most demanding challenge will be the combination with and the development of data structures that enable reloading and unloading afforded feature points for the weighting.

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

This work has partly been supported by the European Commission under Grant no. H2020-ICT-645403-ROBDREAM and the Bavarian Research Foundation under Grant no. AZ-1104-15.

## References

- [1] C. Rink, Z.-C. Marton, D. Seth, T. Bodenmüller, and M. Suppa, "Feature based particle filter registration of 3D surface models and its application in robotics," in *Proceedings of the 26th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '13)*, pp. 3187–3194, Tokyo, Japan, November 2013.
- [2] C. Rink, S. Kriegel, J. Hasse, and Z. Marton, "Onthey particle filter registration for laser data," in *Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR '16)*, Cluj-Napoca, Romania, 2016.
- [3] C. Rink and S. Kriegel, "Streaming Monte Carlo pose estimation for autonomous object modeling," in *Proceedings of the 13th Conference on Computer and Robot Vision (CRV '16)*, Victoria, Canada, 2016.
- [4] S. Kriegel, C. Rink, T. Bodenmüller, and M. Suppa, "Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects," *Journal of Real-Time Image Processing*, vol. 10, no. 4, pp. 611–631, 2015.
- [5] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *The IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [6] J. Sturm, W. Burgard, and D. Cremers, "Evaluating egomotion and structure-from-motion approaches using the TUM RGB-D benchmark," in *Proceedings of the Workshop on Color-Depth Camera Fusion in Robotics (IROS '12)*, October 2012.
- [7] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations," *International Journal of Robotics Research*, vol. 31, no. 12, pp. 1377–1393, 2012.
- [8] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA '13)*, pp. 3748–3754, Karlsruhe, Germany, May 2013.
- [9] T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. McDonald, "Robust real-time visual odometry for dense RGB-D mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '13)*, pp. 5724–5731, Karlsruhe, Germany, May 2013.
- [10] C. Choi and H. I. Christensen, "RGB-D object tracking: a particle filter approach on GPU," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1084–1091, Tokyo, Japan, November 2014.
- [11] K. H. Strobl, *A flexible approach to close-range 3-d modeling [M.S. dissertation]*, Technische Universität München, München, Germany, 2014.
- [12] E. Mair, K. H. Strobl, T. Bodenmüller, M. Suppa, and D. Burschka, "Real-time image-based localization for hand-held 3D-modeling," *Künstliche Intelligenz*, vol. 24, no. 3, pp. 207–214, 2010.

- [13] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Computer Vision—ECCV '96: 4th European Conference on Computer Vision Cambridge, UK, April 15–18, 1996 Proceedings, Volume I*, vol. 1064 of *Lecture Notes in Computer Science*, pp. 343–356, Springer, Berlin, Germany, 1996.
- [14] W. Sepp, S. Fuchs, and G. Hirzinger, "Hierarchical featureless tracking for position-based 6-DoF visual servoing," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 4310–4315, IEEE, Beijing, China, October 2006.
- [15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, Cambridge, Mass, USA, 2005.
- [16] V. Ferrari, F. Jurie, and C. Schmid, "From images to shape models for object detection," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 284–303, 2010.
- [17] 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.
- [18] C.-S. Chen and Y.-P. Hung, "RANSAC-based DARCES: a new approach to fast automatic registration of partially overlapping range images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1229–1234, 1999.
- [19] S. Winkelbach, "Efficient methods for solving 3D-Puzzle-Problems," *it-Information Technology*, vol. 50, no. 3/2009, pp. 199–201, 2008.
- [20] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: efficient and robust 3D object recognition," in *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 998–1005, San Francisco, Calif, USA, June 2010.
- [21] U. Hillenbrand, "Consistent parameter clustering: definition and analysis," *Pattern Recognition Letters*, vol. 28, no. 9, pp. 1112–1122, 2007.
- [22] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 3212–3217, Kobe, Japan, May 2009.
- [23] A. Aldoma, Z.-C. Marton, F. Tombari et al., "Tutorial: point cloud library: three-dimensional object recognition and 6 dof pose estimation," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 80–91, 2012.
- [24] R. B. Rusu, N. Blodow, Z. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '08)*, Acropolis Convention Center, Nice, France, September 2008.
- [25] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust global registration," in *Proceedings of the 3rd Eurographics Symposium on Geometry Processing (SGP '05)*, M. Desbrun and H. Pottmann, Eds., pp. 197–206, Eurographics Association, 2005.
- [26] P. Li, P. Cheng, M. A. Sutton, and S. R. McNeill, "Three-dimensional point cloud registration by matching surface features with relaxation labeling method," *Experimental Mechanics*, vol. 45, no. 1, pp. 71–82, 2005.
- [27] G. Barequet and M. Sharir, "Partial surface and volume matching in three dimensions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 9, pp. 929–948, 1997.
- [28] G. Barequet and M. Sharir, "Partial surface matching by using directed footprints," *Computational Geometry: Theory and Applications*, vol. 12, no. 1-2, pp. 45–62, 1999.
- [29] F. Tombari and L. Di Stefano, "Hough voting for 3D object recognition under occlusion and clutter," *IPSI Transactions on Computer Vision and Applications*, vol. 4, pp. 20–29, 2012.
- [30] J. Glover, R. Rusu, and G. Bradski, "Monte Carlo pose estimation with quaternion kernels and the Bingham distribution," in *Proceedings of the Robotics: Science and Systems Conference*, Los Angeles, Calif, USA, June 2011.
- [31] J. Glover and S. Popovic, "Bingham procrustean alignment for object detection in clutter," in *Proceedings of the 26th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '13)*, pp. 2158–2165, Tokyo, Japan, November 2013.
- [32] T. Bodenmüller, *Streaming surface reconstruction from real time 3D measurements [M.S. thesis]*, Technische Universität München, Munich, Germany, 2009.
- [33] W. R. Scott, G. Roth, and J.-F. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Computing Surveys*, vol. 35, no. 1, pp. 64–96, 2003.
- [34] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: a survey of recent developments," *International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.
- [35] M. Karaszewski, R. Sitnik, and E. Bunsch, "On-line, collision-free positioning of a scanner during fully automated three-dimensional measurement of cultural heritage objects," *Robotics and Autonomous Systems*, vol. 60, no. 9, pp. 1205–1219, 2012.
- [36] S. Khalfoui, R. Seulin, Y. Fougerolle, and D. Fofi, "An efficient method for fully automatic 3D digitization of unknown objects," *Computers in Industry*, vol. 64, no. 9, pp. 1152–1160, 2013.
- [37] L. Torabi and K. Gupta, "An autonomous six-DOF eye-in-hand system for in situ 3D object modeling," *International Journal of Robotics Research*, vol. 31, no. 1, pp. 82–100, 2012.
- [38] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid, "View/state planning for three-dimensional object reconstruction under uncertainty," *Autonomous Robots*, 2015.
- [39] U. Thomas, S. Kriegel, and M. Suppa, "Fusing color and geometry information for understanding cluttered scenes," in *Proceedings of the International Conference on Intelligent Robots and Systems IROS: Robots in Clutter Workshop (IROS '14)*, Chicago, Ill, USA, September 2014.
- [40] K. Bae and D. D. Lichti, "Automated registration of unorganized point clouds from terrestrial laser scanners," in *International Archives of Photogrammetry and Remote Sensing*, vol. 35 of *Proceedings of ISPRS Working Group V/2*, pp. 222–227, 2004.
- [41] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *Proceedings of the IEEE Visualization (VIS '02)*, pp. 163–170, Boston, Mass, USA, November 2002.
- [42] M. Pauly, R. Keiser, and M. H. Gross, "Multi-scale feature extraction on point-sampled surfaces," *Computer Graphics Forum*, vol. 22, no. 3, pp. 281–289, 2003.
- [43] S. Gumhold, X. Wang, and R. Macleod, "Feature extraction from point clouds," in *Proceedings of the 10th International Meshing Roundtable*, pp. 293–305, October 2001.
- [44] E. W. Weisstein, *CRC Concise Encyclopedia of Mathematics*, Chapman & Hall/CRC, 2nd edition, 2002.
- [45] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: representing robot capabilities," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 3229–3236, San Diego, Calif, USA, November 2007.

- [46] J. C. Mitchell, "Sampling rotation groups by successive orthogonal images," *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 525–547, 2008.
- [47] A. Yershova, S. Jain, S. M. Lavalle, and J. C. Mitchell, "Generating uniform incremental grids on  $SO_3$  using the hopf fibration," *International Journal of Robotics Research*, vol. 29, no. 7, pp. 801–812, 2010.
- [48] J. Arvo, "Fast random rotation matrices," in *Graphics Gems III*, D. Kirk, Ed., pp. 117–120, Academic Press Professional, San Diego, Calif, USA, 1992.
- [49] K. Shoemake, "Uniform random rotations," in *Graphics Gems III*, D. Kirk, Ed., pp. 124–132, Academic Press Professional, San Diego, Calif, USA, 1992.
- [50] W. Feiten, P. Atwal, R. Eidenberger, and T. Grundmann, "6D pose uncertainty in robotic perception," in *Advances in Robotics Research*, pp. 89–98, Springer, Berlin, Germany, 2009.
- [51] G. H. Givens and J. A. Hoeting, *Computational Statistics*, Wiley, Hoboken, NJ, USA, 2005.
- [52] J. L. Hintze and R. D. Nelson, "Violin plots: a box plot-density trace synergism," *The American Statistician*, vol. 52, no. 2, pp. 181–184, 1998.
- [53] M. Suppa, S. Kielhöfer, J. Langwald, F. Hacker, K. H. Strobl, and G. Hirzinger, "The 3D-modeller: a multi-purpose vision platform," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 781–787, Rome, Italy, April 2007.
- [54] P. Kremer, T. Wimb, J. Artigas et al., "Multimodal telepresent control of DLR's rollin' justin," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 1601–1602, Kobe, Japan, May 2009.
- [55] T. Bodenmüller, W. Sepp, M. Suppa, and G. Hirzinger, "Tackling multi-sensory 3D data acquisition and fusion," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 2180–2185, San Diego, Calif, USA, November 2007.
- [56] S. Kriegel, *Autonomous 3D modeling of unknown objects for active scene exploration [Ph.D. thesis]*, Technische Universität München (TUM), 2015.



## Research Article

# Underwater Object Tracking Using Sonar and USBL Measurements

**Filip Mandić, Ivor Rendulić, Nikola Mišković, and Đula Nađ**

*University of Zagreb Faculty of Electrical Engineering and Computing, Laboratory for Underwater Systems and Technologies (LABUST), Unska 3, 10000 Zagreb, Croatia*

Correspondence should be addressed to Filip Mandić; [filip.mandic@fer.hr](mailto:filip.mandic@fer.hr)

Received 24 March 2016; Revised 4 July 2016; Accepted 12 July 2016

Academic Editor: Youcef Mezouar

Copyright © 2016 Filip Mandić 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 the scenario where an underwater vehicle tracks an underwater target, reliable estimation of the target position is required. While USBL measurements provide target position measurements at low but regular update rate, multibeam sonar imagery gives high precision measurements but in a limited field of view. This paper describes the development of the tracking filter that fuses USBL and processed sonar image measurements for tracking underwater targets for the purpose of obtaining reliable tracking estimates at steady rate, even in cases when either sonar or USBL measurements are not available or are faulty. The proposed algorithms significantly increase safety in scenarios where underwater vehicle has to maneuver in close vicinity to human diver who emits air bubbles that can deteriorate tracking performance. In addition to the tracking filter development, special attention is devoted to adaptation of the region of interest within the sonar image by using tracking filter covariance transformation for the purpose of improving detection and avoiding false sonar measurements. Developed algorithms are tested on real experimental data obtained in field conditions. Statistical analysis shows superior performance of the proposed filter compared to conventional tracking using pure USBL or sonar measurements.

## 1. Introduction

Tracking underwater targets presents a great challenge in marine robotics due to absence of global positioning signals that are usually available in areas reachable by satellites. In order to tackle this problem, acoustic based sensors such as LBL (long-baseline), SBL (short-baseline), and USBL (ultrashort-baseline) are used for underwater localization and navigation, by triangulating responses obtained from acoustic beacons. While LBLs require inconvenient deploying of underwater beacons around the operational area, USBLs that enable relative underwater localization using acoustic propagation are most often used for tracking underwater objects. The greatest advantage of USBL systems is their easy deployment (the system consists only of two nodes, a transmitter and a transducer) and relatively long range. On the other hand, the precision of USBLs deteriorates with distance and multipath issues may arise. In addition to that, due to acoustic wave propagation, measurements are sparse (arriving at intervals measured in seconds) and time

is delayed depending on the distance between the receiving and the transmitting node.

Besides using USBL devices, multibeam sonar devices (also known as acoustic cameras) are commonly used underwater in order to get relative position measurements. While state-of-the-art multibeam sonars provide almost real-time acoustic image at high frequency with high precision, they are characterized with limited field of view and usually lower range. Unlike USBLs, sonars require additional acoustic image processing in order to obtain position of an object within the field of view, which can often result in false measurements due to noise.

The objective of work presented in this paper is to exploit the advantages of both USBL and sonar devices by fusing their measurements for the purpose of achieving precise and reliable underwater object tracking. The main contributions of this paper are

- (i) development of the tracking filter that fuses USBL and processed sonar image measurements with diverse characteristics, for the purpose of obtaining reliable

tracking estimates at steady rate, even in cases when either sonar or USBL measurements are not available or are faulty;

- (ii) adaptation of the region of interest within the sonar image by using tracking filter covariance transformation for the purpose of improving detection and avoiding false sonar measurements;
- (iii) experimental validation (in field conditions) of the developed tracking algorithms together with comparative analysis that demonstrates the quality of the obtained results.

The main motivation for the presented work arises from the FP7 “CADDY-Cognitive Autonomous Diving Buddy” project that has the main objective to develop a multicomponent marine robotic system comprising of an autonomous underwater vehicle (AUV) and an autonomous surface marine platform that will enable cooperation between robots and human divers. Three main functionalities of the envisioned system include “buddy slave” that assists divers during underwater activities, “buddy guide” that guides the diver to the point of interest, and “buddy observer” that monitors the diver at all times by keeping at a safe distance from the diver and anticipating any problems that the diver may experience.

In the context of the CADDY project one of the main prerequisites for executing envisioned control algorithms and ensuring diver safety during human-robot interaction is precise diver position estimation. In order to achieve this requirement, multibeam sonar imaging is used. However, the main problem that arises when using multibeam sonars is limited field of view. If the observed target (diver or an underwater vehicle) would leave the sonar’s field of view, it would be impossible to track it or even distinguish the tracked object from another target that might enter the field of view. To cope with this problem, fusion between USBL and sonar measurements is incorporated. The low precision USBL measurements are used by the estimator to provide target position, albeit with higher variance. This information is used by the sonar target detector to set the region of interest in which the target is located. Finally, if the sonar detector finds the target in this region of interest, estimator is updated with the high precision (low variance) sonar measurement. The combination of the two sources of measurements ensures reliable target tracking.

The USBL is usually used in vehicle localization and navigation, with a very limited number of papers dealing with target tracking. Fusion of USBL measurements with inertial sensors data and/or vehicle dynamics, used for accurate vehicle localization, is shown in [1, 2]. In [3] the authors have used USBL to track white sharks with an autonomous underwater vehicle, and in [4] USBL tracking was used to track the diver with an autonomous surface vehicle.

Several papers have been published on the use of imaging sonars for object detection and tracking. A method based on the particle filter, shown in [5], was proposed to resolve the problem of target tracking in forward-looking sonar image sequences. In [6] image processing algorithms as well as the tracking algorithms used to take the imaging sonar data and track a nonstationary underwater object are presented. In [7]

the real-time sonar data flow collected by multibeam sonar is expressed as an image and preprocessed by the system. According to the characteristics of sonar images, an improved method has been carried out to detect the object combining with the contour detection algorithm, with which the foreground object can be separated from background successfully. Then the object is tracked by a particle filter tracking method based on multifeature adaptive fusion. In [8] the authors explore the use of such a sonar to detect and track obstacles. In [9] authors provide algorithms for detection of man-made objects on sea floor, where they mostly focus on target-seabed separation issue. The most similar attempt to our work was done in [10], where the sonar was used to detect a human diver. The authors used a similar image processing approach as us, followed by a hidden Markov model-based algorithm for candidate classification.

The papers mentioned above are also mostly focused on the use of image processing and contour based algorithms to detect object. However, they are not directly comparable to our approach as they focus more on the detection part inside the sonar image. Our approach differs from all of the above as it is based on fusion of sonar and USBL. This allows target tracking even when the target is outside of the sonar’s very narrow field of view. It also helps eliminate false positive detections which would cause tracking of the wrong object if multiple objects are present.

The rest of the paper is organized as follows: Section 2 describes deployed sonar image processing algorithms. In Section 3 tracking filter kinematic model is defined. Section 4 gives insight into region of interest adaptation by using transformed position covariance matrix. Experimental results are given in Section 5. The paper is concluded with Section 6.

## 2. Sonar Image Processing

In order to determine target position within the sonar field of view, the sonar image has to be processed. This section is devoted to the description of algorithms used to detect the object in the multibeam sonar image and determine its position within the sonar image.

*2.1. Multibeam Sonars.* Multibeam sonars are also known as acoustic cameras because they, like a video camera, produce a two-dimensional image, although with very different geometric principle. They emit a number of acoustic beams, each one formed to cover a certain horizontal and vertical angle.

*2.2. Target Detection.* Some of the most widely used methods and algorithms for object detection and recognition in images are Haar cascades [11], histograms of oriented gradients [12], and, especially recently, artificial neural networks [13, 14]. Even though these are commonly used in video imagery, they have limited application in sonar-based target detection mostly due to the fact that sonar imagery is usually of very low quality, with incomplete target visualization, preventing even a human observer to reliably detect or recognize the target. In addition to that, our tests with OpenCV implementations

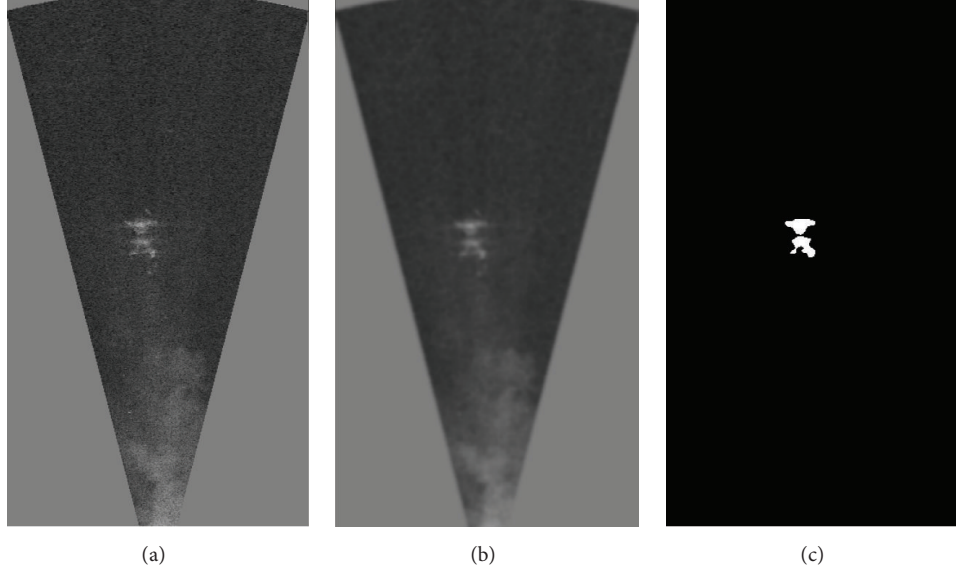


FIGURE 1: First step in sonar image processing demonstrated on an image with a diver in the field of view. (a) Original sonar image; (b) image after blurring; and (c) image after binarization.

of feature descriptors have shown that conventional image descriptors are highly susceptible to noise in sonar image, thus giving poor results.

Due to these reasons, the implemented target detection algorithm relies on clustering contours and finding the ones that are most likely to belong to the target. In order to increase reliability of object detection in sonar image, only the region of interest (ROI) obtained by USBL measurements is searched.

The tracking algorithm implemented can be split into three steps. The first step involves basic image processing, blurring, and binarization of the image. The second step is finding the contours in the obtained binarized image and clustering them together. The final step includes searching for the best candidate inside the region of interest.

**2.2.1. Step 1: Image Processing.** In the first step, a Gaussian blur filter is applied to the image to remove the noise in the image. Often the image is very noisy and has many very little white contours consisting of only a few pixels which we want to ignore. Gaussian blurring is performed by convolving the image with a 2-dimensional Gaussian function:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}. \quad (1)$$

A similar result could be obtained by eroding and dilating the white areas after binarization, as performed in [10]. After blurring, binarization of the image is performed with adaptive thresholding. Each pixel is compared to the mean value of its neighbouring pixels and is set to *white* if it is above that value, or *black* otherwise. Equation (2) describes the binarization algorithm, where  $v_{\text{before}}$  is the pixel value

between 0 and 255 before applying binarization and  $v_{\text{after}}$  takes the value of either 0 or 255 after binarization:

$$v_{\text{after}}(x, y) = \begin{cases} 255 & \text{if } v_{\text{before}} > T(x, y) \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where

$$T(x, y) = \frac{1}{4M+2} \sum_{i=-M}^M \sum_{j=-M}^M v_{\text{before}}(x+i, y+j). \quad (3)$$

The results of image blurring and binarization are displayed in Figure 1.

**2.2.2. Step 2: Contour Detection and Clustering.** In the second step, all white contours in the image are clustered together if they are closer than some predefined distance. This distance is chosen depending on the target tracked. For example, if a human diver is tracked, we can expect that the diver's head or limbs appear disjoint from the torso. To cluster them together, it is reasonable to allow contours that are closer than half a meter to be clustered together.

To achieve the clustering, a graph approach could be taken by using Kruskal's minimum spanning tree algorithm with early termination. However, simple union find algorithm with disjoint set data structure can achieve the same with even lower complexity: while Kruskal's algorithm runs in  $O(E \log V)$ , where  $E$  is the number of edges in the graph and  $V$  is the number of vertices, union find runs in  $O(n\alpha(n))$ , where  $n$  is the number of items and  $\alpha(n)$  is the extremely slow-growing inverse of the Ackermann function [15].

The results of the implemented clustering algorithm are displayed in Figure 2. The diver's body is disconnected, but with the clustering algorithm the pieces are merged together into the same cluster and marked with the same color.

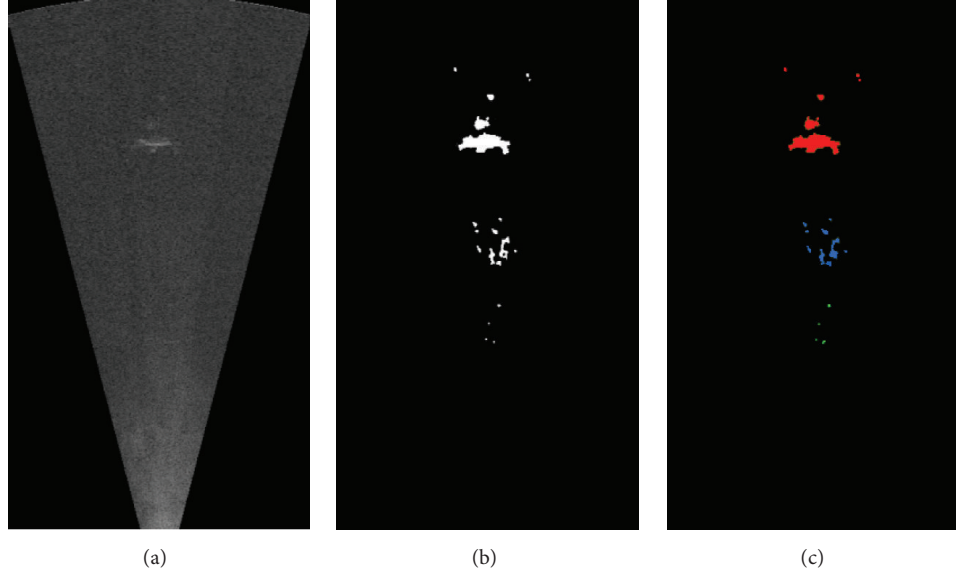


FIGURE 2: (a) Original sonar image; (b) image after binarization; and (c) contours clustered into three clusters.

**2.2.3. Step 3: Finding Target inside the ROI.** The final steps assumes that the approximate area where the target should be already familiar; that is, it is estimated by an extended Kalman filter that uses USBL measurements and sonar measurements from the previous step, as explained in the following chapter. This assumption is required due to the fact that accurate tracking using only sonar image is difficult, especially if there are other similar objects present in the image, for example multiple divers or autonomous underwater vehicles.

All the clusters that are inside the ROI are given a quality score based on a criterion that consists of two parts:

- (1) Distance from the ROI center: the closer the cluster to the ROI center is, the higher its score is.
- (2) Visual similarity of each cluster and the target: even though very little training data is available, similarity of the cluster is compared with known target's properties (by comparing the size and shape and applying a simple template-based object detector or a small neural network).

The object with the highest score above the (empirically set) threshold is then selected as the most likely target. This allows us to score multiple objects and reliably choose the one that fits best both the current estimated position of the target (obtained from the tracking filter) and the known characteristics of the target.

### 3. Tracking Filter

Once the target position within the sonar field of view is known, it can be used as a measurement for the tracking filter. In order to estimate underwater target position from available measurements, extended Kalman filter (EKF) is deployed. Only kinematic model is used for target position estimation since target dynamics are usually unknown. Equations for

the vehicle's translatory motion are given with (4) where  $\mathbf{p} = [x \ y \ z]^T$  is the position vector and  $\psi$  is the orientation of the vehicle in the earth-fixed coordinate frame. Input  $\mathbf{v} = [u \ v \ w]^T$  is speed vector and input  $r$  is orientation rate in body-fixed coordinate frame:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{R}(\psi) \mathbf{v}, \\ \dot{\psi} &= r. \end{aligned} \quad (4)$$

Rotation matrix  $\mathbf{R}(\psi)$  is given with

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

The vehicle tracking the underwater target and carrying the imaging sonar is modeled as an overactuated marine surface vehicle; that is, it can move in any direction by modifying the surge, sway, and heave speed, while attaining arbitrary orientation in the horizontal plane. Kinematic model of the target is given with the following set of equations:

$$\begin{aligned} \dot{\mathbf{p}}_B &= \begin{bmatrix} \dot{x}_B \\ \dot{y}_B \\ \dot{z}_B \end{bmatrix} = \begin{bmatrix} u_B \cos \alpha_B \\ u_B \sin \alpha_B \\ w_B \end{bmatrix} + \boldsymbol{\xi}_{pb}, \\ \dot{\mathbf{v}}_B &= \begin{bmatrix} \dot{u}_B \\ \dot{w}_B \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \boldsymbol{\xi}_{vb}, \\ \dot{\alpha}_B &= r_B + \xi_{\alpha}, \\ \dot{r}_B &= \xi_{rb}, \end{aligned} \quad (6)$$

where  $\mathbf{p}_B$  is target position vector and  $\mathbf{v}_B$  is speed vector consisting of surge speed  $u_B$  and heave speed  $w_B$ . State  $\alpha_B$



denotes target course and  $r_B$  course rate. Process noise for respective states is denoted by  $\xi$ . Finally, state vector of target absolute position tracking filter is

$$\mathbf{x} = [\mathbf{p}^T \ \psi \ \mathbf{v}_B^T \ \mathbf{p}_B^T \ r_B \ \alpha_B]^T, \quad (7)$$

where subscript  $B$  denotes target related states. Measurement vector is given with

$$\mathbf{y} = [\mathbf{p}_m^T \ \psi_m \ z_{Bm} \ r_{mUSBL} \ \Theta_{mUSBL} \ r_{ms} \ \Theta_{ms}]^T. \quad (8)$$

Vector  $\mathbf{p}_m$  denotes vehicle position measurement,  $\psi_m$  heading measurement, and  $z_{Bm}$  target depth measurement, while  $r_{m(\cdot)}$  and  $\Theta_{m(\cdot)}$  denote USBL and sonar range and bearing measurements, where respective measurement equations are

$$r_m = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} + \nu_r, \quad (9)$$

$$\Theta_m = \arctan(\Delta y, \Delta x) - \psi + \nu_\Theta. \quad (10)$$

Parameter  $\nu$  denotes measurement noise which is, in this case, modeled as zero mean Gaussian noise. Note that bearing measurement is relative; therefore, there is a heading state  $\psi$  included in (10).

The target depth measurement  $z_{Bm}$  can be acquired using elevation angle and range measurements between two units provided by the USBL device. Also, acoustic communication can be used to transmit depth measurements taken directly on board the target if they are available.

It was already noted that sonar measurements arrive with high frequency and small delay while USBL measurements are low frequency and delayed; therefore, Kalman filter measurement matrix  $\mathbf{H}$  is changed every time step, according to available measurements. Also, to account for measurement delays methods of backward recalculation can be applied.

## 4. Region of Interest Adaptation

In order to improve detection and avoid false sonar measurements, region of interest (ROI) is defined by using tracking filter estimates covariance. Sonar image processing can be performed in relative Cartesian or polar coordinates; therefore, it is necessary to transform absolute position covariance accordingly.

**4.1. Covariance Transformation.** By definition, covariance matrix of vehicle and target relative position can be written as

$$\Sigma = \mathbf{E} \left[ (\mathbf{p}_p - \mathbf{E}(\mathbf{p}_p)) (\mathbf{p}_p - \mathbf{E}(\mathbf{p}_p))^T \right], \quad (11)$$

where  $\mathbf{p}_p = [\Delta x \ \Delta y]^T$ . The assumption is that the position of the vehicle carrying the sonar is known without uncertainty and that all uncertainty stems from unknown target position. The assumption is made that the vehicle and the target are at the same depth when the target is visible in the sonar image, since sonar vertical field of view is quite small. For this reason, target depth is considered to be known and is omitted from  $\mathbf{p}_p$ .

**4.1.1. Covariance Transformation between Two Cartesian Coordinate Systems.** Covariance transformation between relative position in earth-fixed NED coordinate frame and relative position in body-fixed frame is given with (12) where  $\Sigma$  is NED coordinate covariance matrix and  $\mathbf{R}_p$  is the rotation matrix given with (13) [16]:

$$\Sigma_{\text{rel}} = \mathbf{R}_p \Sigma \mathbf{R}_p^T, \quad (12)$$

$$\mathbf{R}_p = \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{bmatrix}. \quad (13)$$

**4.1.2. Covariance Transformation between Cartesian and Polar Coordinate Systems.** Relationship between relative Cartesian and polar coordinate system is given with the nonlinear equation expression:

$$\begin{bmatrix} r \\ \Theta \end{bmatrix} = \begin{bmatrix} \sqrt{\Delta x_{\text{rel}}^2 + \Delta y_{\text{rel}}^2} \\ \arctan(\Delta y_{\text{rel}}, \Delta x_{\text{rel}}) \end{bmatrix}. \quad (14)$$

In order to transform the covariance matrix, Jacobian of Cartesian-to-polar covariance transformation is written as [17]

$$\mathbf{J} = \begin{bmatrix} \frac{\partial r}{\partial \Delta x_{\text{rel}}} & \frac{\partial r}{\partial \Delta y_{\text{rel}}} \\ \frac{\partial \Theta}{\partial \Delta x_{\text{rel}}} & \frac{\partial \Theta}{\partial \Delta y_{\text{rel}}} \end{bmatrix} = \begin{bmatrix} \frac{\Delta x_{\text{rel}}}{r} & \frac{\Delta y_{\text{rel}}}{r} \\ -\frac{\Delta y_{\text{rel}}}{r^2} & \frac{\Delta x_{\text{rel}}}{r^2} \end{bmatrix}. \quad (15)$$

Finally, covariance matrix in relative polar coordinates  $\Sigma_{\text{pol}}$  is calculated as

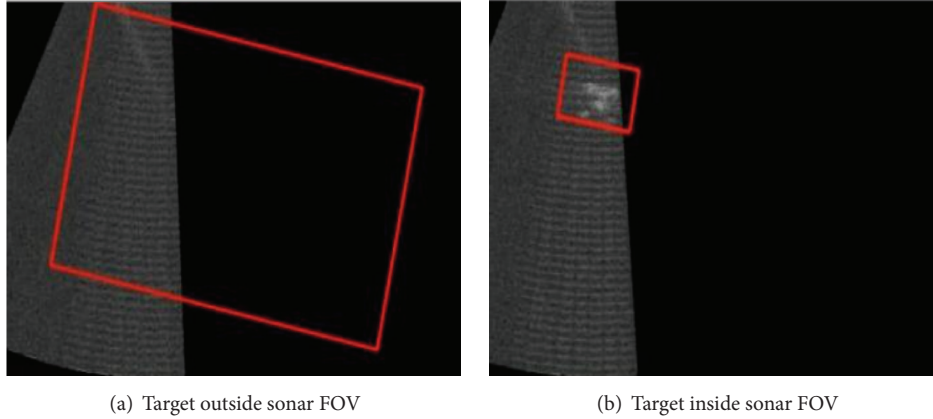
$$\Sigma_{\text{pol}} = \mathbf{J} \Sigma_{\text{rel}} \mathbf{J}^T. \quad (16)$$

**4.2. Using the Tracking Filter Covariance for Region of Interest.** After transforming the filter covariance in relative coordinate frames (Cartesian or polar), it is used to define a region of interest used in sonar tracking as described in Section 2. More specifically, given covariances  $D_x$  and  $D_y$  in relative coordinate frames, estimated object size along these axes  $S_x$  and  $S_y$ , and tracking filter estimate position  $(T_x, T_y)$ , region of interest is defined as follows:

$$\begin{aligned} \text{ROI}_x &= \left[ T_x - \frac{S_x}{2} - 3\sqrt{D_x}, T_x + \frac{S_x}{2} + 3\sqrt{D_x} \right], \\ \text{ROI}_y &= \left[ T_y - \frac{S_y}{2} - 3\sqrt{D_y}, T_y + \frac{S_y}{2} + 3\sqrt{D_y} \right], \\ \text{ROI} &= \text{ROI}_x \times \text{ROI}_y, \end{aligned} \quad (17)$$

where covariances  $D_x$  and  $D_y$  are members  $\Sigma_{\text{rel}1,1}$  and  $\Sigma_{\text{rel}2,2}$  from relative covariance matrix (12). Similarly, in case of polar coordinates, line segments are defined for radius  $r$  and angle  $\phi$ , and the region of interest is the Cartesian product between the two.

Figure 3 illustrates the size of the region of interest and the estimated location of the target (center of the ROI). Figure 3(a) shows the case when only USBL measurements



(a) Target outside sonar FOV

(b) Target inside sonar FOV

FIGURE 3: Sonar image with region of interest (ROI).

are available, while Figure 3(b) shows results with both USBL and sonar measurements. The ROI (covariance) is much smaller when sonar measurements are available. However, it is worth noting that tracking is possible even when the target is outside of sonar field of view, due to the fact that USBL measurements are fused within the tracking filter.

Minimum area of the ROI can be set by adjusting measurement noise variance  $\gamma$ , while the rate of ROI growth when there are no measurements available can be defined by adjusting process noise parameters, especially  $\xi_{pb}$ .

## 5. Experimental Results

**5.1. Experimental Setup.** Experiments related to target tracking using sonar and USBL data were conducted in October 2015 in Biograd na Moru, Croatia, during CADDY project validation trials. The experimental setup consisted of an autonomous underwater vehicle BUDDY AUV and an autonomous overactuated marine surface platform PlaDyPos, both developed in the Laboratory for Underwater Systems and Technologies [4, 18]. Multibeam sonar was installed horizontally and forward-looking on the BUDDY AUV here referred to as the vehicle, while PlaDyPos vehicle played the role of the target to be tracked. Buddy AUV, shown in Figure 6, has been developed in the scope of CADDY project. It is equipped with six thrusters that allow omnidirectional motion in the horizontal plane, thus ensuring decoupled heading and position control. Among other sensors, it is equipped with a multibeam sonar and a USBL used for positioning and communication. Overall dimensions of the BUDDY AUV are  $1220 \times 700 \times 750$  mm and the weight is about 70 kg. PlaDyPos vehicle, used as a target, is a small scale overactuated unmanned surface marine vehicle capable of omnidirectional motion. It is equipped with four thrusters in “X” configuration. This configuration enables motion in the horizontal plane under any orientation. The vehicle is 0.35 m high and 0.707 m wide and long and weighs approximately 25 kg.

The sonar used for experiments reported in this paper is Soundmetrics ARIS 3000 [19], with 128 beams, covering  $30^\circ$

angle in horizontal and  $14^\circ$  in vertical plane. It supports two operating modes: high frequency at 3 MHz for higher detail at ranges up to 5 meters and low frequency at 1.8 MHz for ranges up to 15 meters. Also, during experiments, Seatrec X150 and X110 USBL modem pair was used [20]. The combined modem/USBL units are designed as a very compact assembly. They operate in the frequency band 24–32 kHz and the communication rate of 100 bps can be achieved.

USBL modems were installed on both the vehicle and the target object. During experiments, it was assumed that the vehicle and the target are in the same horizontal plane when the target is visible in the sonar image; that is, the vehicle and the target have the same depth. Filtered GPS measurements, from the measurement units installed aboard the vehicle and the target, are taken as ground truth. It should be noted that errors in ground truth measurements are present due to inherent GPS measurement covariance and the fact that different GPS modules were installed on the vehicle and the target, which induced small variable drift. By visual inspection of sonar images it was observed that when image processing algorithm detects correct target, acquired relative sonar measurements are more accurate and precise than relative distance calculated from GPS measurements.

**5.2. Results.** During validation trials, a large number of target tracking experiments were conducted. In this paper, the analysis of results is performed on two datasets, each describing one experimental scenario. In Scenario 1, the vehicle is moving while the target is static or slowly drifting (Figure 4). In Scenario 2, the vehicle is static while the target is moving (Figure 5). In both scenarios, three different filter configurations are investigated, defined by available measurements: (i) “Sonar” configuration where only sonar measurements are available, (ii) “USBL” configuration where only USBL measurements are used, and, finally, (iii) “Sonar + USBL” configuration where both sonar and USBL measurements are available.

The dataset corresponding to Scenario 1 is shown in Figure 4, while Figure 5 shows the dataset of Scenario 2. In both figures, first two subplots show north and east

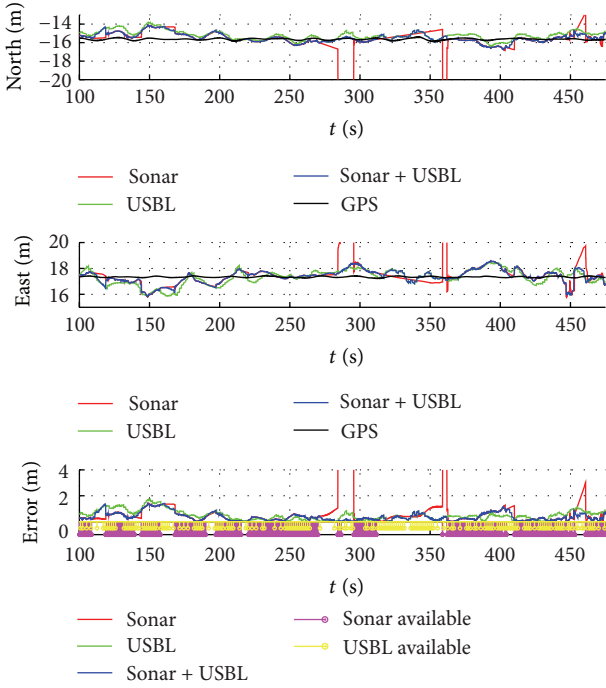


FIGURE 4: Scenario 1: vehicle moving, stationary target.

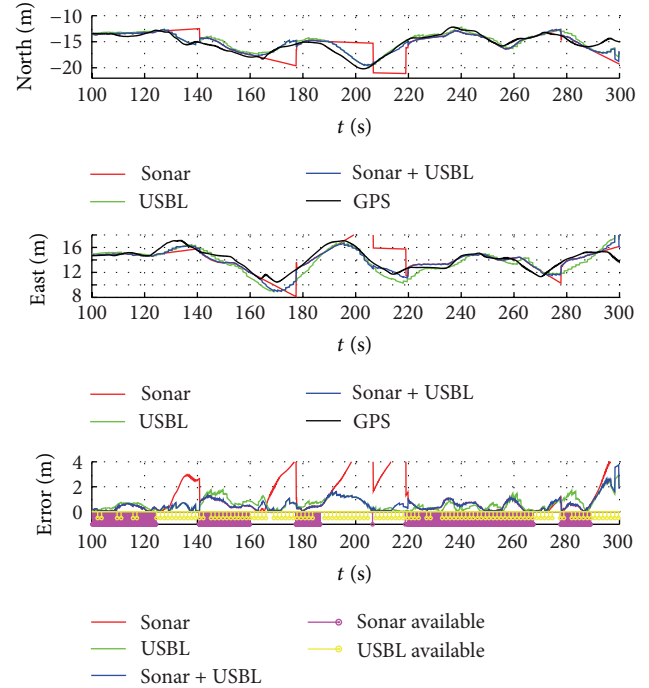


FIGURE 5: Scenario 2: stationary vehicle, moving target.

TABLE 1: Distance error comparison.

Scenario	Sonar availability [%]	USBL availability [%]
(1) Vehicle moving, target static	31.7	4.5
(2) Vehicle static, target moving	28.0	5.7

coordinates, while the third subplot shows the errors (Euclidean distance) between the estimated positions and the ground truth obtained via GPS measurements from both the vehicle and the target. Red line shows the results obtained from the tracking filter that uses only sonar measurements (filter configuration “Sonar”), green line is obtained from tracking filter that uses only USBL measurements (filter configuration “USBL”), and the blue line is the results obtained from the tracking filter that utilizes both sources of measurements as they become available (filter configuration “Sonar + USBL”). Black line shows the ground truth position.

**5.2.1. Frequency of Measurements.** In the third subplot of both Figures 4 and 5, one can appreciate magenta and yellow circles that mark the time instances in which sonar and USBL measurements were available. Table 1 gives a comprehensive analysis on the amount of time when sonar and USBL measurements were available. Taking into account that the tracking filter provides estimates at 10 Hz sampling frequency, it can be seen that, in both scenarios, sonar measurements were available at around 30% of sampling instances, whether due to lower running frequency of

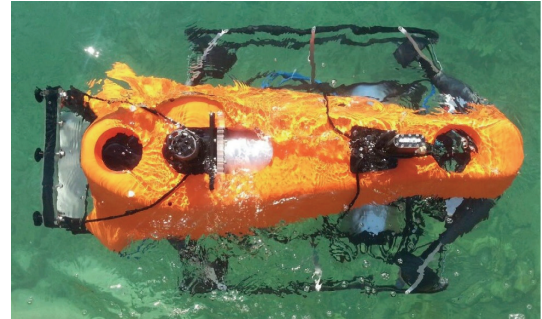


FIGURE 6: BUDDY AUV in water seen from above. The front end has a waterproof casing with a tablet.

the sonar image processing algorithms or due to the fact that some of the time the target was not present in the sonar image. On the other hand, USBL measurements are available at only 5% of time instances. It can be seen from Figures 4 and 5 that USBL measurement availability is consistent during the whole duration of both scenarios; however, the update rate of USBL measurements is around 2 s which corresponds to approximately 5% availability taking into consideration the 10 Hz tracking filter sampling frequency.

**5.2.2. Comparison of Filter Configurations.** Datasets shown in Figures 4 and 5 instantly show the disadvantage of filter configuration “Sonar”—whenever sonar measurements are not available, the position estimate drifts from the true value. One can appreciate this more clearly in Figure 7(a) which shows a 45-second segment of the full-time response. The

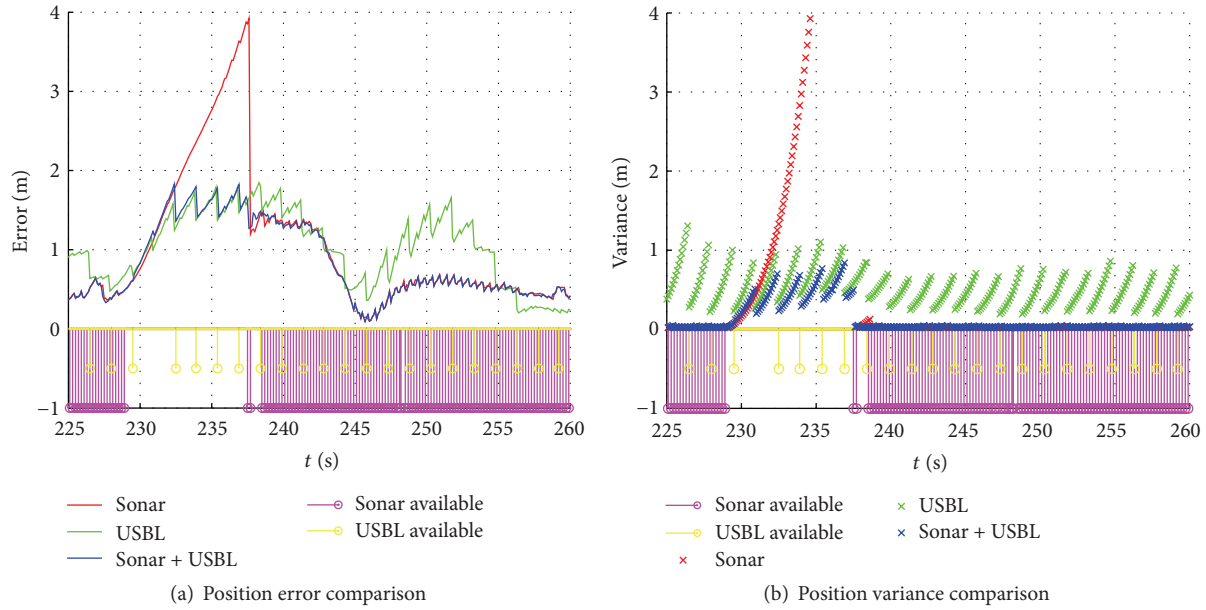


FIGURE 7: Tracking filter data inset.

fact that position estimates quickly drift when the target is not in the sonar field of view can have serious consequences, especially in situations where a human diver is the target to be tracked and the position estimate is used to control the vehicle position relative to the diver.

On the other, using only USBL measurements (as in filter configuration “USBL”) enables tracking even when the target is outside field of view, as long as there is a clear path between the target and the vehicle, ensuring obstruction-free propagation of the acoustic wave. However, USBL measurements arrive at a low update frequency.

Fusion of sonar and USBL measurements combines the best features of both types of measurements: high precision of sonar measurements and availability of USBL measurements. This is also clear from Figure 7(b) which shows tracking filter position variance for each filter configuration. Using both USBL and sonar measurements, filter estimated variance is more stable regardless of which measurements are available. In the case when only USBL measurements are used, variance grows between two measurements. In the case when only sonar measurements are used, variance grows unboundedly when measurements are not available.

**5.2.3. Statistical Analysis of Results.** In order to quantify the result that the sonar and USBL fusion approach gives, the most reliable results metrics is defined based on the localization error obtained as Euclidean distance between the ground truth position (obtained using GPS on board both the vehicle and the target) and position estimates using all three filter configurations. These errors are shown in the form of a boxplot, where Figure 8(a) gives the analysis for Scenario 1 (shown in Figure 4), and Figure 8(b) gives analysis for Scenario 2 (shown in Figure 5). Both boxplots show results for filter configurations “Sonar,” “USBL,” and “Sonar

+ USBL.” In addition to that, the results are shown for the filter configuration “Sonar,” taking only into account position estimates when sonar measurements were available, that is, when the target was within the sonar field of view—this is labeled with “Sonar (available).”

As expected, the “Sonar (available)” data gives the most precise results for both scenarios. However, this measure does not represent the real situation, since it was shown that the target was available within the sonar field of view only around 30% of time in both scenarios. This measure should be regarded as the best possible results that can be obtained using the measuring devices available in the setup.

Localization error boxplot for filter configuration “Sonar” over the whole dataset shows that the results are the least precise as it can be seen in Figures 8(a) and 8(b). This is a result of the fact that all the data is included, even the data when target is lost from the sonar FOV and there is no way to estimate target position since the filter presumes that target continues in the direction it was going before leaving sonar FOV.

In both scenarios, filter configuration “USBL” provides the least accurate mean position error, but the variance over the whole dataset is much lower than in the filter configuration “Sonar.”

As it can be seen from Figures 8(a) and 8(b), in both scenarios, filter configuration “Sonar + USBL” gives mean localization error lower than filter configurations “Sonar” and “USBL.” The same can be said for position error variance. It should be noticed that this filter configuration provides results which are very close to our “ideal” situation where the target is always present in the sonar image, that is, the “Sonar (available)” case.

In Scenario 2 (the case of the static target), all the obtained localization error statistical results are smaller but the same



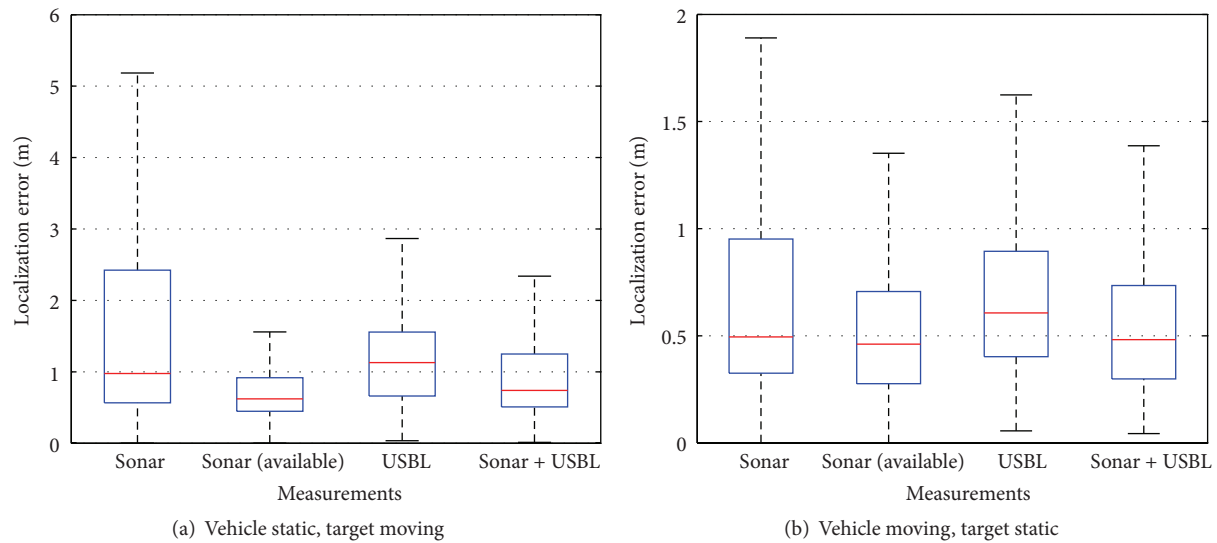


FIGURE 8: Localization error boxplot.

pattern can be observed as in Scenario 1 (the case of the moving target).

**5.2.4. Video.** Video representing the results with target position estimate obtained by fusing sonar and USBL measurements can be found in [21].

## 6. Conclusions

The paper addresses the issue of underwater target tracking by using sonar and USBL measurements. The results that were used to analyze the tracking quality were obtained from data gathered using BUDDY AUV, an autonomous underwater vehicle developed for diver-robot interaction that served as the tracking vehicle in the experiments, and PlaDyPos autonomous surface marine platform that played the role of the target to be tracked.

The experiments have shown that sonar measurements, when available, are very accurate and precise, but there is always a possibility of detecting false targets especially in cluttered environments. Also, when tracking divers false measurements due to bubbles are common. Using USBL measurements even when the target is in the sonar FOV helps reduce number of false detection incidents. For example, in Figure 4 we can see false detection at time instants 280 s, 360 s, and 450 s. Using USBL and sonar sensor fusion discards such measurements since they are out of ROI, and there are no abrupt changes of position estimate. As a consequence, mean localization error is the lowest as seen in Figure 4. Finally, the developed tracking filter that fuses USBL measurements with position measurements obtained from the processed sonar image shows superior performance.

Future work will focus on exploiting knowledge gained through these experiments for designing algorithms in which underwater vehicle actively tracks the underwater target while trying to keep it in the sonar FOV as often as possible.

## Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work is supported by the European Commission under the FP7-ICT project “CADDY-Cognitive Autonomous Diving Buddy” under Grant Agreement no. 611373. Filip Mandić is financed by the Croatian Science Foundation through the project for the young researcher career development.

## References

- [1] M. Morgado, P. Oliveira, C. Silvestre, and J. F. Vasconcelos, “Embedded vehicle dynamics aiding for USBL/INS underwater navigation system,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 1, pp. 322–330, 2014.
- [2] A. Jayasiri, R. G. Gosine, G. K. I. Mann, and P. McGuire, “AUV-based plume tracking: a simulation study,” *Journal of Control Science and Engineering*, vol. 2016, Article ID 1764527, 15 pages, 2016.
- [3] G. E. Packard, A. Kukulya, T. Austin et al., “Continuous autonomous tracking and imaging of white sharks and basking sharks using a remus-100 auv,” in *Proceedings of the IEEE OCEANS*, pp. 1–5, San Diego, Calif, USA, September 2013.
- [4] N. Stilinic, D. Nad, and N. Miskovic, “Auv for diver assistance and safety—design and implementation,” in *Proceedings of the OCEANS*, pp. 1–4, Geneva, Switzerland, May 2015.
- [5] T. Zhang, W. Zeng, L. Wan, and S. Ma, “Underwater target tracking based on Gaussian particle filter in looking forward sonar images,” *Journal of Computational Information Systems*, vol. 6, no. 14, pp. 4801–4810, 2010.

- [6] D. W. Krout, W. Kooiman, G. Okopal, and E. Hanusa, "Object tracking with imaging sonar," in *Proceedings of the 15th International Conference on Information Fusion (FUSION '12)*, pp. 2400–2405, IEEE, Singapore, September 2012.
- [7] M. Li, H. Ji, X. Wang, L. Weng, and Z. Gong, "Underwater object detection and tracking based on multi-beam sonar image processing," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '13)*, pp. 1071–1076, Shenzhen, China, December 2013.
- [8] Y. Petillot, I. T. Ruiz, and D. M. Lane, "Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 2, pp. 240–251, 2001.
- [9] E. Galceran, V. Djapic, M. Carreras, and D. P. Williams, "A real-time underwater object detection algorithm for multi-beam forward looking sonar," *IFAC Proceedings*, vol. 45, pp. 306–311, 2012.
- [10] K. J. DeMarco, M. E. West, and A. M. Howard, "Sonar-based detection and tracking of a diver for underwater human-robot interaction scenarios," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '13)*, pp. 2378–2383, IEEE, Manchester, UK, October 2013.
- [11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. 1-511–1-518, IEEE, Kauai, Hawaii, USA, December 2001.
- [12] 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)*, pp. 886–893, San Diego, Calif, USA, June 2005.
- [13] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.
- [14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, December 2012.
- [15] R. E. Tarjan, "Efficiency of a good but not linear set union algorithm," *Journal of the Association for Computing Machinery*, vol. 22, pp. 215–225, 1975.
- [16] T. Soler and M. Chin, "On transformation of covariance matrices between local cartesian coordinate systems and commutative diagrams," in *Proceedings of the ASP-ACSM Convention*, pp. 393–406, 1985.
- [17] A. J. Haug, *Bayesian Estimation and Tracking: A Practical Guide*, John Wiley & Sons, New York, NY, USA, 2012.
- [18] D. Nad, N. Mišković, and F. Mandić, "Navigation, guidance and control of an overactuated marine surface vehicle," *Annual Reviews in Control*, vol. 40, pp. 172–181, 2015.
- [19] SoundMetrics, "ARIS 3000," <http://www.soundmetrics.com/products/aris-sonars/aris-explorer-3000>.
- [20] J. A. Neasham, G. Goodfellow, and R. Sharpouse, "Development of the 'Seatrak' miniature acoustic modem and USBL positioning units for subsea robotics and diver applications," in *Proceedings of the OCEANS-Genova*, pp. 1–8, IEEE, Genoa, Italy, May 2015.
- [21] Multibeam sonar and USBL fusion for tracking, 2015, <https://www.youtube.com/watch?v=O6ndThfLT-0>.

## Research Article

# Robotic Visual Tracking of Relevant Cues in Underwater Environments with Poor Visibility Conditions

**Alejandro Maldonado-Ramírez and L. Abril Torres-Méndez**

*Robotics and Advanced Manufacturing Group, CINVESTAV Campus Saltillo, 25900 Ramos Arizpe, COAH, Mexico*

Correspondence should be addressed to L. Abril Torres-Méndez; [abriltorresm15@gmail.com](mailto:abriltorresm15@gmail.com)

Received 26 March 2016; Revised 18 June 2016; Accepted 28 June 2016

Academic Editor: Youcef Mezouar

Copyright © 2016 A. Maldonado-Ramírez and L. A. Torres-Méndez. 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.

Using visual sensors for detecting regions of interest in underwater environments is fundamental for many robotic applications. Particularly, for an autonomous exploration task, an underwater vehicle must be guided towards features that are of interest. If the relevant features can be seen from the distance, then smooth control movements of the vehicle are feasible in order to position itself close enough with the final goal of gathering visual quality images. However, it is a challenging task for a robotic system to achieve stable tracking of the same regions since marine environments are unstructured and highly dynamic and usually have poor visibility. In this paper, a framework that robustly detects and tracks regions of interest in real time is presented. We use the chromatic channels of a perceptual uniform color space to detect relevant regions and adapt a visual attention scheme to underwater scenes. For the tracking, we associate with each relevant point superpixel descriptors which are invariant to changes in illumination and shape. The field experiment results have demonstrated that our approach is robust when tested on different visibility conditions and depths in underwater explorations.

## 1. Introduction

Visual tracking of relevant regions in scenes with poor visibility is an important problem in robotic vision research. In particular, for the autonomous robotic exploration of natural underwater structures (e.g., coral reefs), it is fundamental to perform a closer, cautious, and a noninvasive analysis of the changes that occur in the structure of interest to assist in the research of marine biologists. Usually, human intervention is required to indicate which regions are of interest for monitoring by remotely operating the underwater vehicle. As this can be quite demanding, the need of using an Autonomous Underwater Vehicle (AUV) is very appealing. Moreover, the visual and control algorithms need to be quite robust and run in real time in order to be effective. In recent years, several systems capable of collecting information, dynamically or statically, in underwater environments have been developed. In the case of AUVs, great efforts have been made to provide them with sufficient autonomy to perform specific tasks. Thus, the main challenge is to transfer to

the robotic agent the ability of recognizing what regions are of interest for monitoring and to keep those regions on view for a certain period of time to be able to obtain useful visual data for its posterior analysis. However, as these targets or regions of interest may be located far from the vehicle, they need to be detected from the distance. The rapid attenuation of electromagnetic radiation in water limits the range of optical sensors. Also, the existence of variable lighting and the presence of suspended particles (also known as marine snow) cause geometrical and color distortions that result in poor visibility. Furthermore, the structure (in terms of geometric shape) of coral reefs is practically null. Since underwater environments are highly unstructured and constantly changing environments, one of the main problems that still remains open is the accurate estimation of the robot's position and orientation. This makes the detection and tracking of visual cues difficult. Considering the mentioned problems, if the goal is to cautiously explore the fragile marine life that exists in coral reefs, it is necessary to first detect visual targets that are relevant for the exploration and then

robustly track them so that the robot movements are not erratic or abrupt. In other words, the tracking must be stable enough to allow for smooth control movements in the robotic system.

We are interested in allowing an AUV to conduct an exploration of coral reefs according to how a human diver would do it: that is, the route to follow is guided by the features in the environment that catch her attention. It turns out that for underwater environments using this type of exploration there exists limited research work in the literature. For example, in [1], a method is presented to classify the captured images by the robot according to the degree of novelty contained in the features. The novelty parameter is an indicator used to control the speed of the robot along a predefined path. An extension of this work is presented in [2], where the movement of the robot is controlled to be directed to areas in the image with more visual content, causing the robot to move to areas containing coral reef and ignore the areas where only sand is present. One important thing to note is that, in an exploration mode, it is crucial not to limit the movements of the robot to a previous specified path; instead, the approach used should allow for a more natural scanning. In this sense, a diver (sufficiently curious and fearless) exploring a coral reef for the first time will be guided by what catches her attention, despite not having prior information about what she could find.

In this research work, we present a real-time visual-based framework to robustly detect and track relevant features from the distance with the aim of exploring coral reefs. The real-time performance in robotics applications is fundamental since the tracked features will help to direct the exploration trajectories in subsequent captured images while estimating the relative pose of the robot. We build upon our previous work [3, 4]. In [4], a visual attention model, adapted to underwater scenes, was presented for the first time. The inputs were a set of videos taken underwater. Although the visually relevant cues were likely to be detected on subsequent frames, it was not enough to keep track of a particular relevant cue for long. Moreover, it only worked when water conditions were optimal, thus failing when poor visibility conditions were present. In [3], we characterized the colors of relevant features by using a perceptually uniform color space. We compared the CIE *Lab* and the *Lαβ*, which were able to define a super-color-pixel descriptor to describe a relevant region by using its chromatic channels only. The color opponent processing (*blue-yellow* and *green-red*) makes the recovering of color underwater easy, in particular red and yellow tones, by enhancing their contrast wrt the blue/green tonalities of sea waters.

In this paper, we have extended our previous work in many aspects. First, we give a detailed description of each of the stages involved in our Aquatic Visual Attention (AVA) model as well as improvements to have a better saliency map in terms of the compactness of the relevant regions. Second, we have compared the performance of the proposed framework. On one hand, we compare the quality in the detection of regions of interest of our AVA model in underwater scenes at different depths with the classic Neuromorphic

Vision Toolkit method. On the other hand, we compare the robustness of superpixels descriptors for tracking the most relevant region of interest with other methods of object tracking.

The contribution of this paper is a novel computational visual attention model built to work on underwater environments, namely, coral reefs. The proposed visual attention model focuses on detecting as well as tracking relevant regions. The purpose of having a tracker is to lead the motion of an Autonomous Underwater Vehicle (AUV) in an exploration task. This way the AUV should be able to detect, without human intervention or any kind of precise information of a particular region, which part of the coral reef could draw the attention for a human and move towards it.

The outline of the paper is as follows. Section 2 presents background on the perception of color in underwater scenes and also on visual attention models. Section 3 describes our model and its implementation. The experimental results, comparison of the performance of the proposed framework, and discussion are presented in Section 4. Finally, in Section 5, the conclusions and future work are given.

## 2. Background

**2.1. Underwater Perception of Color.** Poor visibility conditions underwater affect the perception of color. This is due to the attenuation of light, water conditions, distance to objects, depth, and other factors [5]. Visibility in foggy days is very similar to that of underwater images. The effect is that near objects are clearer while distant objects gradually disappear. This effect is illustrated in Figure 1 by comparing images of the same natural scene under foggy and normal day conditions. The mountains in the back of Figure 1(a) cannot be seen in Figure 1(b).

Color perception in common sea water diminishes according to the distance or depth where the object of interest is located. In most cases, the color in objects that are more than 10 meters of distance are almost indistinguishable (see Figure 2). As for depth, the first color to disappear is red; beginning as soon as 3 m of depth there is almost not red light left from the sun. From 5 m to 10 m, the range from orange to yellow lights is lost. By 25 m, only blue light remains [5]. Figure 3 shows an example of an image of our AUV at different depth and water conditions. We know that the color of our robot is red by the sides. By verifying the color of the intensity pixels in a small window (zoomed in), we see that the color is very different from red, ranging from dark red to dark blue. However, the processes carries out in our brain adjust the colors up to certain grade.

**2.2. Perceptually Uniform Color Spaces for Color Discrimination.** Natural structures underwater, such as the formations of coral reef, are rich in color and texture. They may have certain shapes, but they do not always follow a specific pattern or geometry. Thus, if we want to have a descriptor for a given feature, the only cue to detect and recognize would be color. In this trend, the discrimination of color is the problem we want to solve. This is different to the color





(a) Image in a sunny day

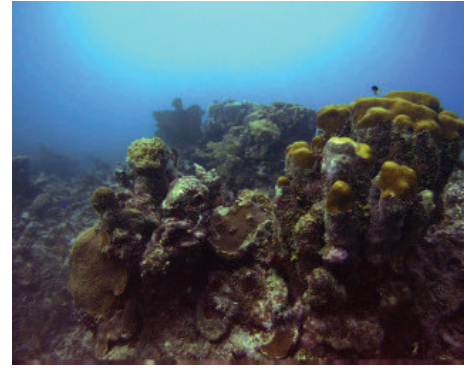


(b) Image in a foggy day

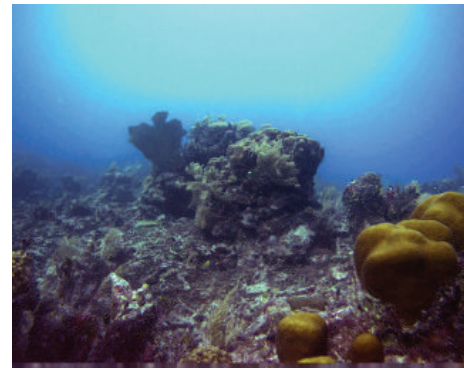
FIGURE 1: The same outdoor natural scene under different weather conditions. In (b), the effects of fog clearly show how objects in the distance gradually disappear (e.g., mountains in the back, clearly seen in (a), are gone). This effect is similar in underwater scenes (see Figure 2). Taken from [3].

restoration problem, in which a good result is basically that with a *natural look* of color appearance, but there is not guarantee that the true original color has been recovered whatsoever.

To discriminate color, one needs to measure the differences among the entire range of visible colors in a way that matches perceptual similarity as good as possible. This task can be simplified by the use of perceptually uniform color spaces, in which a small change of a color will produce the same change in perception anywhere in the color space. This is due to the fact the chromatic channels are spaced further apart. Examples of perceptual uniform color spaces are the CIE *Lab* and the  $L\alpha\beta$ . On one hand, the CIE *Lab* model was specifically developed to describe all the color that the human eye can perceive [6] and it was designed to preserve the perceptual color distance. Thus, the Euclidean distance is an accurate representation of the perceptual color difference. The *a* channel values represent the relative light purplish red (magenta) or greenness of each pixel. Shifting the curve upwards builds up magentas and weakens greens. The *b* channel does the same for yellow versus blue. Altering the slope of these curves changes color contrast, while adjusting parts of the curve selectively changes different ranges of colors. On the other hand, the  $L\alpha\beta$  is a decorrelated principal component color space. This color space was derived from a large ensemble of hyperspectral images of natural scenes using the first-order statistics of the images. Because of its decorrelation property of three



(a)



(b)

FIGURE 2: Examples showing the effect of distance perception in underwater. Same as in foggy days, distant objects gradually disappear. However, an additional effect is that near objects appear bigger than they actually are.

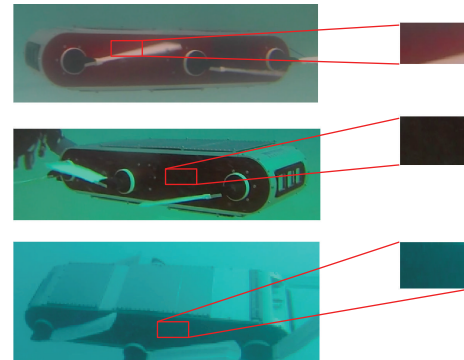


FIGURE 3: Example of color perception at different depth and water conditions.

channels, the  $L\alpha\beta$  space has been used for color mapping in terrestrial applications [7, 8] and just recently it was used for underwater applications for color correction [9] with good results.

**2.3. Experiments: Underwater Color Discrimination.** We carried out experiments to visually compare how color can be discriminated when using the RGB, HSV, CIE *Lab*, and

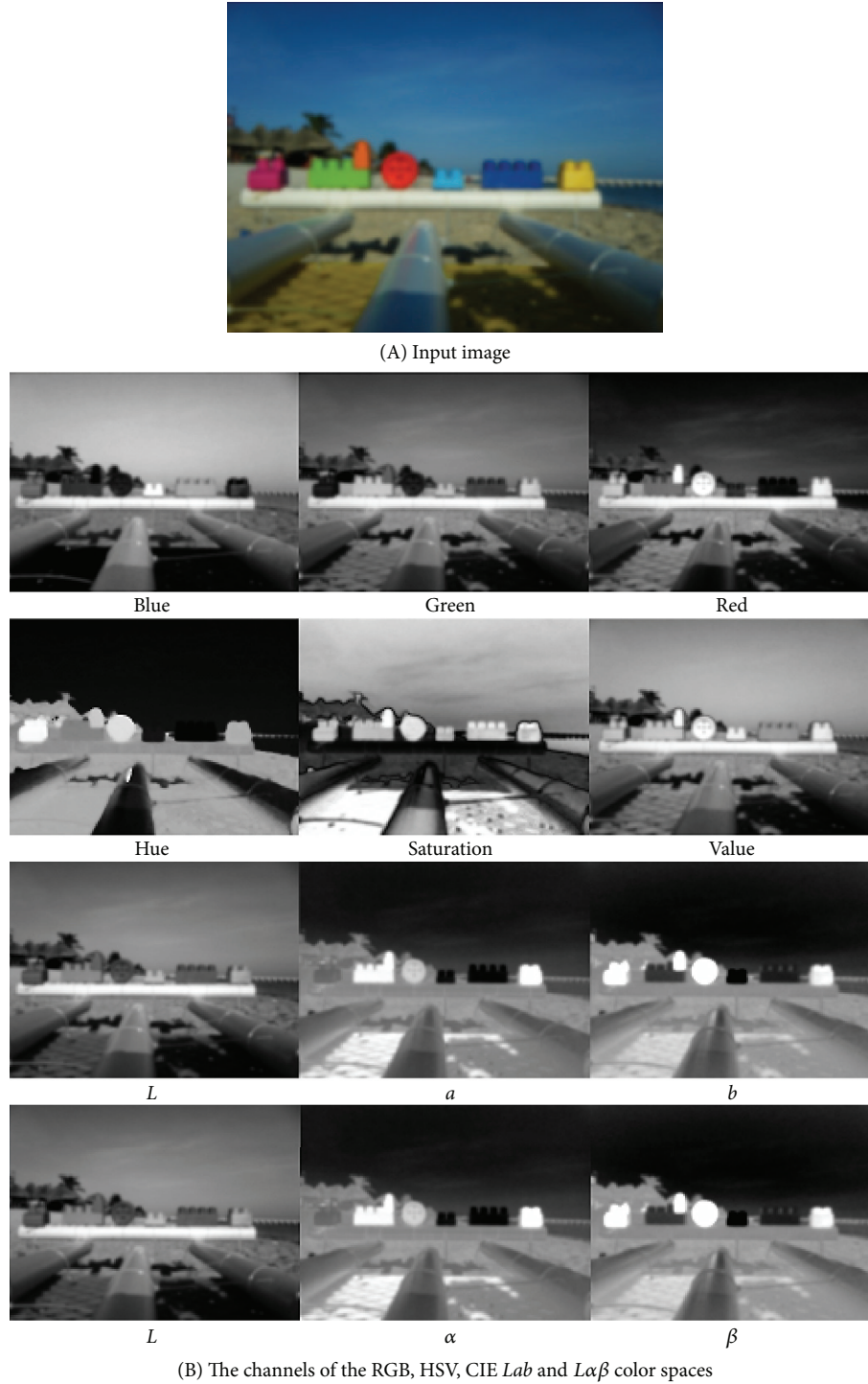


FIGURE 4: Chromatic channels of different color spaces applied to an outdoor scene. Note that the red channel of RGB and the  $a$  channel of CIE  $Lab$  are in the last column in order to visually facilitate the comparison. Taken from [3].

the  $L\alpha\beta$  color spaces. It is important to remind that our goal is to see how red and yellow tonalities are detected. We are neither doing a restoration of the color nor enhancing the color in images. The underwater images were taken on three different sea waters, from the Caribbean and the Yucatan peninsula. As it was previously mentioned, the advantage of using opponent color spaces is because for this type of images;

one of the opponent colors is basically the color of water, that is, a bluish or greenish tone. Since colors are usually defined in terms of human observation, the evaluation of the performance of an algorithm that involves color information is a more qualitative aspect than a quantitative one. Figures 4 and 5 show examples of using different color spaces in an outdoor and underwater images under poor visibility

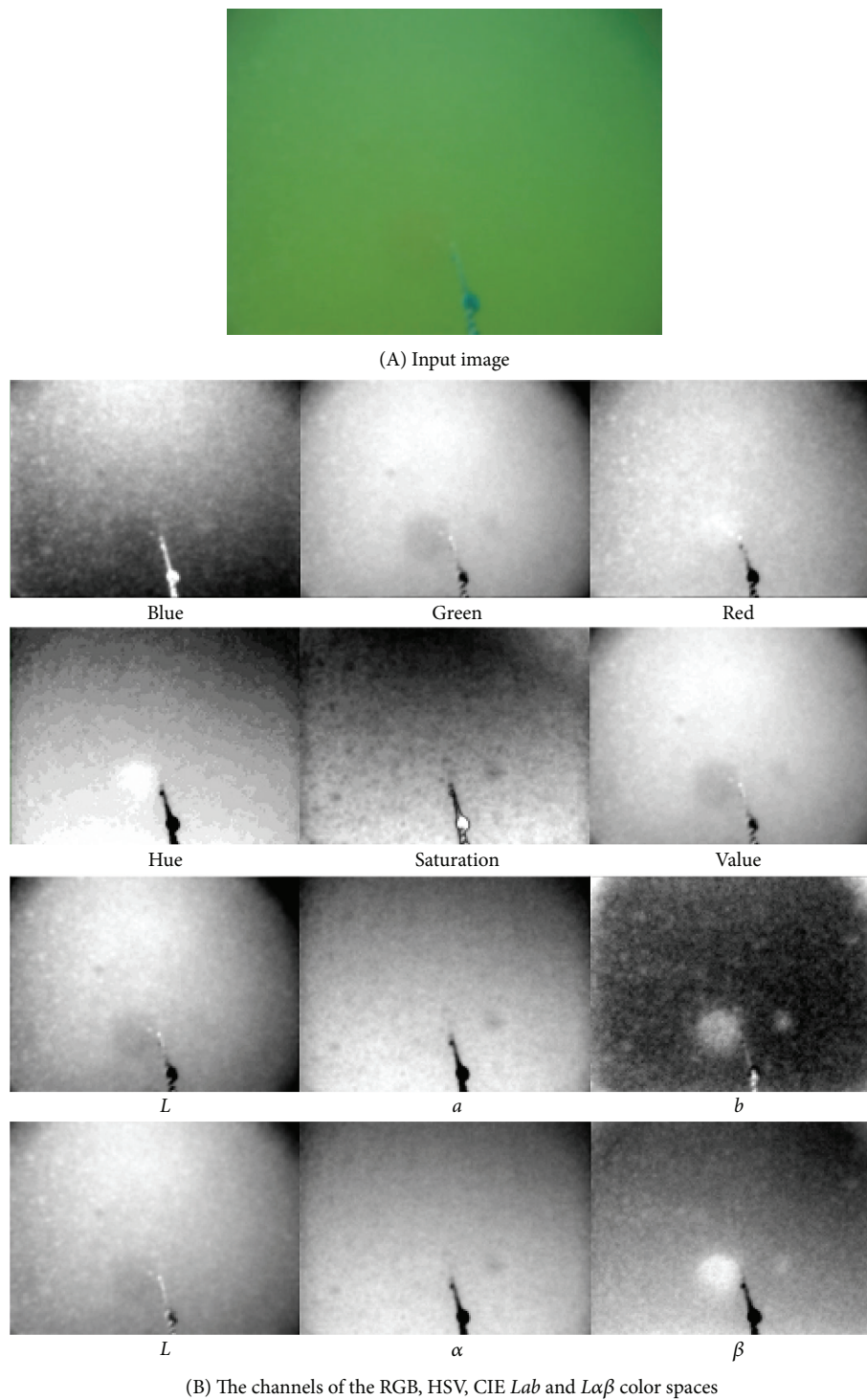


FIGURE 5: Chromatic channels of different color spaces applied to an underwater scene. Note that the red channel of RGB and the  $a$  channel of CIE  $Lab$  are in the last column in order to visually facilitate the comparison. Taken from [3].



conditions which are illustrated. Figure 4 depicts in the first row the input image taken outside water; then in the next rows, the three channels of the RGB, HSV, CIE *Lab*, and  $L\alpha\beta$  color spaces, respectively, are shown. In similar arrangement of images, Figure 5 shows the three channels of each color space when applying to underwater images in poor visibility conditions.

It can be observed that all color spaces discriminate red and yellow colors in the images. However, in underwater, only the CIE *Lab* and  $L\alpha\beta$  color spaces were able to discriminate the red color of the ball. This conclusion arises from a visually qualitative comparison.

**2.4. Visual Attention Models.** Visual attention is a selective process that allows us to determine what draws our attention according to the visual stimuli we receive from our environment. Several works have been done in the area of neuropsychology to understand how humans pay attention to what we see. Even today, there are several theories about how the human visual attention system works. Based on those theories, various computer models have been proposed. Studies about visual attention originally emerged in the area of psychology and neurophysiology over a century ago [10], when scientists began to develop theories and models to explain it. But it was not until 1987, in the work presented by Koch and Ullman [11], when the first model of a biologically inspired computational attention was published. After this work many more were proposed, being the work by Itti et al. [12] the most relevant to date. A comprehensive survey of visual attention and its implementation in computer systems can be found in [13].

One of the motivations for incorporating attention capabilities in systems that process huge amount of information is to reduce the amount of the data to be processed. This can be achieved by taking only the information. In the area of computer vision it is particularly noticeable, as images contain thousands, even millions of pixels. The problem of reducing image information has been addressed in various ways. To mention a few, there exist methods that are based on the detection of points of interest, such as the Harris' corner detector [14], SURF [15], or the well-known SIFT [16]. Also, there are detectors of lines, ellipses, and circles [17, 18]. Another approach that has also been applied involves the predictive methods, which use information regarding the task to be performed to limit the amount of information to be processed.

Two of the more popular attention models, due to their easy implementation, flexibility, and fast computation, are the Neuromorphic Vision Toolkit (NVT) proposed by Itti et al. [12] and the attention system called Visual Object detection with a computational attention system (VOCUS) by Frintrop et al. [19]. The Focus of Attention (FoA) is the place in the image that draws the attention of the system. Itti et al. [12] searched for the FoA by using a Winner-Take-All neural network. Frintrop et al. [19] find the point with the highest saliency value by scanning every point, and the most salient region is determined by seed region growing.

Recently, visual attention models have been used in robotic applications [20], and in underwater applications to primarily assist marine biologists in their review of underwater videos. For example, Walther et al. [21] and Edgington et al. [22] detect objects and potentially interesting visual events for humans in order to label the frames of a video stream as interesting or boring. In both research works, the NVT [12] model is used. The videos used in those works were recorded by a Remotely Operated Vehicle (ROV).

Barat and Rendas [23] present a visual attention system for detection of manufactured objects. Their model is based on the minimum description length test for detecting the motion of contrasting neighboring regions. After that, a statistical technique is adapted to determine the boundary of the object. Correia et al. [24] use intensity, motion, and edge maps as features for their visual attention model to detect the Norway lobsters and help scientist to quantify them.

In all these works, the visual attention models are used for aiding humans in the task of analyzing video streams. In our case, we want the visual attention model to direct the robot motion through the automatic detection and tracking of features that could be of interest for a human during an exploration. Particularly, we are interested in transferring abilities to an AUV in order to detect regions of interest without human supervision while successfully navigating the environment. For the case of autonomous underwater exploration the visual attention algorithm requires real-time performance. Moreover, as hardware limitations in underwater robots are still an issue, the algorithms should have a low computational cost.

### 3. The Proposed Method

In this section, the method we propose for detecting and tracking relevant features in underwater scenes is described. Our approach for detection of relevant features uses some key ideas of Itti's and Frintrop's visual attention models [12, 19]. A computational visual attention algorithm detects relevant regions in an image emulating the human visual attention.

Traditionally, the detection of relevant features relies on a saliency map—a gray-scale image in which the brightest part is the most relevant in terms of features such as intensity, color, and orientation. Given that the existing natural objects in underwater scenes lack specific orientation and shape, our attention model strongly relies on color information. However, the inherent poor visibility and color degradation of sea water are critical at distances and depths greater than 10 meters. For that reason, it is important to select an appropriate color space to achieve an effortlessly underwater image enhancement. We use the CIE *Lab* color space.

The most relevant regions can be found by selecting the location with the highest value in the saliency map. In a sequence of underwater images of the same scene, it is common that the location associated with the highest value of saliency changes drastically from one frame to the next. This is due to the variations in the illumination and/or local water conditions. Thus, if the location of the region of interest in the image domain is going to lead the motion of the vehicle



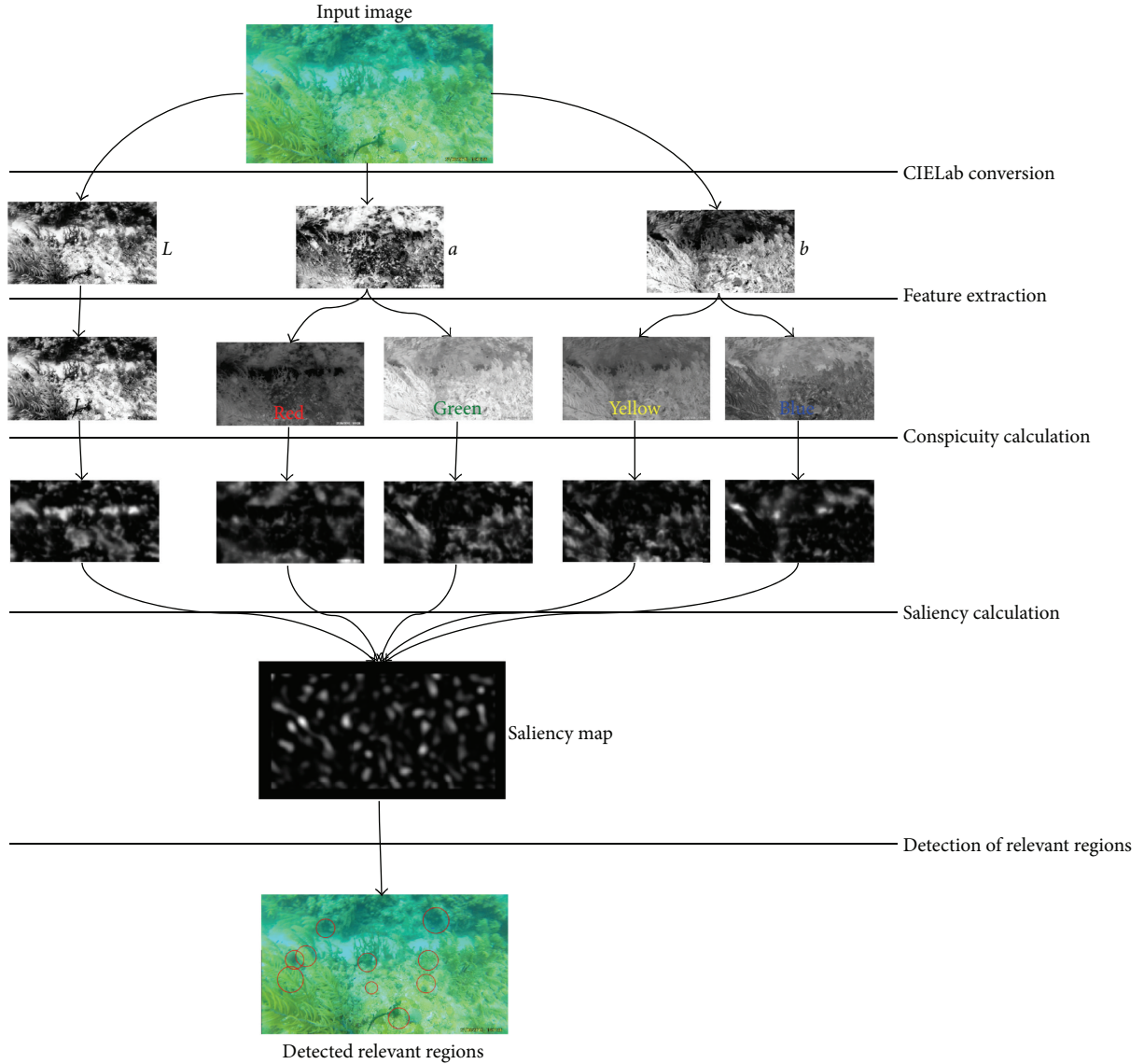


FIGURE 6: A general overview of the proposed method for detecting relevant regions.

in the space domain, then a robust tracking of the same or very similar region (in position and appearance) is crucial to minimize the erratic motion of the vehicle.

In the following sections, we describe in more detail each of the steps involved in our visual attention model. In Figure 6, a general overview of the proposed method for detecting relevant regions is depicted.

**3.1. Preprocessing of the Image.** The input image is scaled to a proper size (typically 0.25 of the original size). Then, the image is converted to the CIELab color space. In Section 2.3 some advantages of this color space as well as some examples can be found.

**3.2. Getting the Features Maps.** We use intensity and color (red, yellow, green, and blue) as features. The intensity map corresponds to  $L$ -channel of the CIELab image. The colors

are extracted from  $a$  and  $b$  channels, as described in [25], as follows:

$$F_i(x, y) = V_{\max} - \|ab(x, y) - p\|, \quad (1)$$

where  $F_i(x, y)$  is  $i$ th feature map,  $V_{\max} = 255$  in 8-bit depth images,  $p = (a_d, b_d)$  is the desired color to extract in terms of the chromatic channels, and  $ab(x, y)$  is the  $ab$ -channel of the image. The color feature maps are gray-scale images in which the intensity indicates how near is the desired color to the original color of the pixel. We do not use the orientation feature in our model, as it mainly works well in structured environments (e.g., man-made environments).

**3.3. Getting the Conspicuity Maps.** The conspicuity map is a gray-scale image where the most relevant regions (in terms of a feature) appear brighter than other regions. The first step to calculate these maps is to build a Gaussian pyramid for each

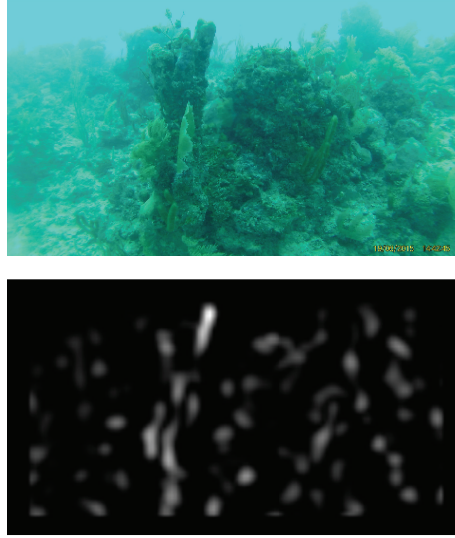


FIGURE 7: Example of the saliency map obtained from an underwater image.

feature. A Gaussian pyramid is built by applying a Gaussian filter and then downsampling the image in half. If we apply this process again to the resulting image, we can construct the other levels of the pyramid. The number of levels used in the pyramid depends on the size of the input image and the size of the relevant regions to be found. Bigger regions require more level in the pyramid to be effectively detected. We use a 5-level pyramid: that is, five scales  $s_m = \{1, 0.5, 0.25, 0.125, 0.0625\}$ .

An important aspect to consider in any computational visual attention system is highlighting the relevant part for each feature map. This is usually done by using a center-surround mechanism (also called *center-surround difference*), which is inspired in cells of the human visual receptive field [26]. In our approach, these differences are implemented as convolution. Let  $\mathbf{P}(d)$  be the image in the level  $d$  of the pyramid for a given feature, then the center-surround differences are applied as follows:

$$\mathbf{P}'(d, \sigma) = \mathbf{P}(d) - \mathbf{K}(\sigma) * \mathbf{P}(d), \quad (2)$$

$$\mathbf{K}(\sigma) = \frac{1}{(2\sigma + 1)^2 - 1} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & 0 & \vdots \\ 1 & \cdots & 1 \end{bmatrix}_{(2\sigma+1) \times (2\sigma+1)},$$

where  $\sigma$  defines the size of the mask and  $*$  is the convolution operator between an image and the mask. For each level of the pyramid two maps are obtained,  $\mathbf{P}'(d, 3)$  and  $\mathbf{P}'(d, 4)$ .

The resulting images from the application of the center-surround differences are resized to 0.25 of the size of the original image. After that, all the images from the same feature pyramid are added to a single image  $\mathbf{C}$ , called conspicuity map.

It is important to note that, contrary to [12, 25], in which the created conspicuity map involves all colors, we calculate a conspicuity map for *each* of the color features. This allows

us, in the posterior stages, to indicate which colors have more relevance during the exploration.

**3.4. Getting the Saliency Map.** The saliency map is a gray-scale image, in which the most relevant parts appear brighter. To obtain this map, a Difference of Gaussians (DoG) is applied to each conspicuity map. After that, a weighted sum of the resulting maps (normalized in the range  $[0, 1]$ ) is computed. Formally, the saliency map is calculated as follows:

$$\mathbf{S} = \sum_i w_i \cdot \text{DoG}(\mathbf{C}_i), \quad (3)$$

where index  $i$  represents each of the conspicuity maps obtained from each feature. By assigning different weight values  $w_i$  to each map, we can give a preference to a particular color tonality. The weighted sum can be seen as a simple way to incorporate a top-down attention. Unlike VOCUS, in which a training image containing the object to search is used, our model does not need images of a particular object. In any case, we just need to have some information about the possible dominant color of an object of interest. An example of a saliency map can be seen in Figure 7.

**3.5. Searching of Relevant Points.** Once the saliency map is calculated, a search for  $q$  more relevant points or *regions of interest* (RoI) is carried on. As in VOCUS, a sequential search of the highest values over all image pixels is done. Also, to avoid repeating the location of points, we apply an inhibition of return approach. This way, the area surrounding each of the relevant points is inhibited and the next relevant point will be far from the previous one, allowing for a sparse distribution of relevant regions. Figure 8 shows an example of the RoIs detected in an image. Unlike our previous work [3, 4] where a fixed area around a given point is inhibited, in this work a Seeded Region Growing method [27] is used over the saliency map to determine a circle that encloses the area to be inhibited.



FIGURE 8: Example of relevant regions detected (enclosed by red circles) in an underwater scene.

**3.6. Superpixel-Based Descriptors for Tracking of Relevant Regions.** From the set of regions of interest detected with the AVA algorithm, the *Focus of Attention* (FoA) is the one with the highest value. Thus, the FoA represents the region that caught the attention the most in an underwater scene.

For some applications, once a FoA is selected, it is important to keep track of it in the following images in a sequence. As our purpose is to explore an underwater environment, our AVA model must keep track of the same (or very similar) FoA in subsequent frames as much as possible, if and only if this region is still among the most relevant ones. We are interested in this behavior because it will lead the actions of a robot during an exploration task. Having abrupt changes of the FoA's location from one frame to the next one may cause an erratic motion.

To track a region or a point in an image, a descriptor is needed. We propose to use a superpixel-based descriptor. A particular advantage that superpixels offer is that they adapt their shape to enclose similar characteristics of a region, in terms of color and position. Thus, if we associate with each relevant region to be tracked the superpixel characteristics they belong to, we are assuring a local robust description.

The procedure is as follows. The input image is segmented in  $M$  superpixels using the SLIC algorithm [28] with  $M \ll N$ , where  $N$  the number of pixels in the input image. Each superpixel is a set of pixels with similar features and it is characterized by a 5-dimensional vector of the form  $[L_s, a_s, b_s, x_s, y_s]$ , where  $L, a, b$  are the mean color values of the pixels belonging to a given superpixel in the CIELab color space and  $(x_s, y_s)$  is the centroid of the superpixel. A relevant region is described by the vector  $\mathbf{s}$  composed from the components  $a_s, b_s, x_s$ , and  $y_s$  from the superpixels it belongs to. It can be noted that  $L_s$  component is not taken into account because the illumination in this kind of environments can change from frame to frame.

Once we have the descriptors for each of  $q$  most relevant regions, we choose the closest one (the most similar) to the descriptor of the FoA from the previous frame. The chosen region becomes the FoA of the current frame. The distance (similarity) measure between two superpixel-based descriptors,  $\mathbf{s}_j$  and  $\mathbf{s}_k$ , is based on the SSD metric as in [28], without the luminance part:

$$D(\mathbf{s}_j, \mathbf{s}_k) = \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2}, \quad (4)$$

where

$$\begin{aligned} d_c &= \sqrt{(a_j - a_k)^2 + (b_j - b_k)^2}, \\ d_s &= \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2}, \end{aligned} \quad (5)$$

where  $N_c$  and  $N_s$  are normalization factors for the distance in the color and image space, respectively. These values were set as described in [29].

Figure 9 illustrates the use of superpixels to achieve a stable tracking of similar FoAs in a region of interest. If the distance from the closest saliency descriptor to the previous FoA descriptor is greater than a defined threshold  $\mu$ , the distances are ignored and the point with the highest saliency value is chosen as the new FoA.

## 4. Experimental Results

In this section, we present the experimental results to validate the parts of the proposed approach. First, we show the outcome of the comparison of detected relevant regions by humans and the proposed system. Then we compare the relevant regions detected by our approach (AVA) and the Neuromorphic Visual Toolkit (NVT) [12]. After that, a comparison in terms of tracking is shown. Finally, we present the outcome of using the proposed approach to guide the motion of an underwater robot in an exploration task.

**4.1. Relevant Regions Detected by Humans.** A comparison between the regions considered as relevant by a group of people and by the proposed approach is presented. The purpose of this experiment is to show that our visual attention algorithm is able to detect regions that have the potential to draw the attention of a human. Thus, the AUV can autonomously explore the underwater environment in terms of what a human could consider relevant.

We asked 32 people (16 men and 16 women between 20 and 30 years of age with no experience in coral reefs) to select (by clicking on the screen) the region that attracts their attention the most in a set of underwater images containing various scenes of coral reef. Then, we applied our algorithm on the same set of images. Two regions are considered coincident if their circles of radius  $r$  centered at the relevant region present an overlapping greater than 80%. Figure 10

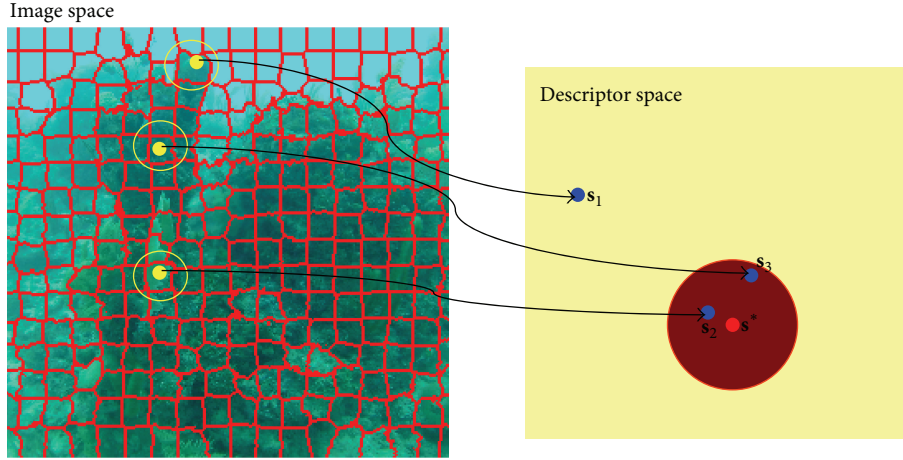


FIGURE 9: Finding the next Focus of Attention. The descriptor for each detected relevant region is obtained. The next Focus of Attention is the closest descriptor to the previous FoA's descriptor  $s^*$ . For a descriptor  $s_j$  to be considered as a FoA candidate its distance to  $s^*$  should be less than a given threshold  $\mu$  (represented as the circle around  $s^*$ ).





















Image 1	Image 2	Image 3	Image 4
 0	 13%	 66%	 63%
 0	 60%	 16%	 0
 0	 10%	 0	 0
 0	 10%	 0	 0
 93%	 0	 0	 0

FIGURE 10: Regions of interest (RoI) detected by our method and the percentage of coincidence made by a group of 32 people. Each column shows patches containing the five most relevant regions detected by our system in the corresponding image. More than half of the group choose as relevant at least one of the areas detected with our visual attention system.

depicts the obtained results. Each row of the array of images in the figure contains the five most relevant regions detected by AVA and the percentage of people that considered the same region as relevant.

In the presented results, more than half of the group choose as relevant at least one of the areas detected with our visual attention system. This study shows us that our model approximates the way a person will select regions of interest in coral reefs environments. This is important since we want our robot to explore the coral reef as a diver visiting it for the first time.

**4.2. Comparison of Detected Regions.** In order to measure the performance of our method in terms of detecting relevant regions on underwater scenes, we carried on an analytical

comparison of our results with those obtained using the NVT method [12]. This method was used as implemented in the Saliency Tool Box (STB) (the STB can be found in <http://www.saliencytoolbox.net/>) [30]. For the STB, the default configuration was used. The features used by our algorithm are the intensity and color (red, green, yellow, and blue). For our method, we set the weights of all the conspicuity maps equal to 1.

For this study, we need to determine if the relevant regions detected by the computational attention methods can be considered of interest for a human. This can be done by using a person's judgement. However, this criterion can be very subjective and time consuming for a large set of images. We decided to simplify the evaluation and assumed that the interesting regions should appear on parts of the coral reef: that is, the areas that visually correspond only to water are not considered of interest. First, to divide the image into water and nonwater regions, we applied an adapted version of the robust superpixel-based classifier proposed in [31].

This classifier is used to segment the floor in indoor environments for mobile robot reactive navigation. We have adapted this classifier so it can segment water instead. One of the advantages is that it can be trained online with the current water conditions, and once it is running, it can automatically adapt to possible changes in tonality. All this makes the classifier quite robust. A classification example is depicted in Figure 11.

To perform the comparison test, both algorithms were set to detect the five most relevant regions on each of the 1550 frames in six video sequences. The videos contain a great variety of water conditions, depths, and scenarios of the coral reef of Costa Maya, Mexico. It is important to mention that many of the images in the sequence present challenging situations, for example, high brightness from the sun, bluish and greenish tonalities in case of images taken at deeper locations, and blurriness due to camera motion. All the detected regions that fell into the nonwater area were counted as relevant. In Table 1, the results obtained are shown.



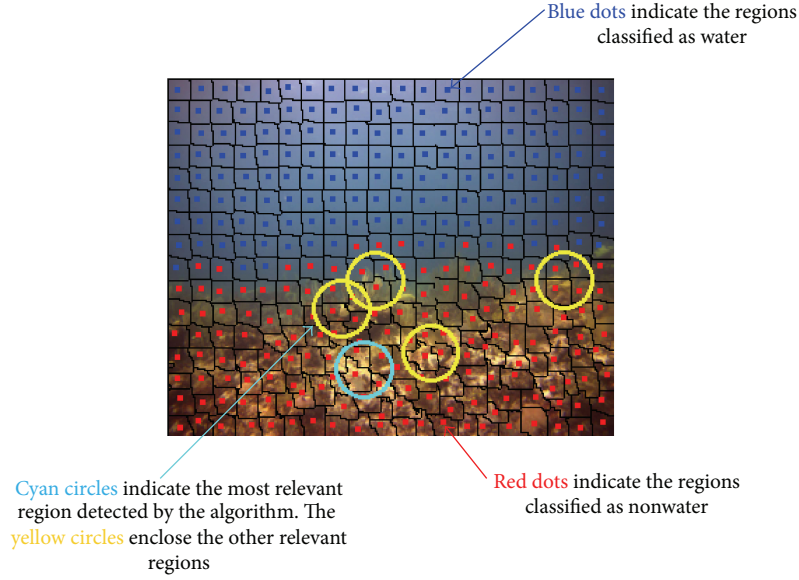


FIGURE 11: Example of a classification of water (blue dots superpixels) and nonwater (red dots superpixels) regions. Also in this image, five relevant regions detected by our visual attention algorithm are shown. The most relevant region is enclosed in a cyan circle whereas the rest are enclosed in yellow circles.

It can be seen that the percentage of regions detected as interesting by our method is greater than the percentage when using NVT. Let us not forget that these results are over the five most relevant regions detected by both algorithms. We carried on another test, in which only the first most relevant regions were considered. If this relevant region was considered as interesting then it was counted as correct. Table 2 shows the percentages of the interesting regions for the two algorithms for comparison purposes. Although the proposed algorithm percentage is higher than the NVT, the difference is minimal. For this case, however, we have noted (by visually inspecting the detected regions) that many of the relevant points detected by the NVT method were on areas containing only sand or rock formations of brown or black color, which are not considered of interest in an exploration task.

In Figure 12, some images from the video sequences with the relevant regions detected by the algorithms are shown. Qualitatively, in terms of relevance, it can be seen that some of the regions detected by the NVT algorithm are on water or on irrelevant parts like sand or shadows. Also, it can be noted, in the sixth row of both figures, that the regions detected by our algorithm tend to be in the coral reef despite of the abrupt illumination changes due to the sun.

As was shown in Tables 1 and 2, the detected regions by our algorithm tend to be part of the coral reef in more occasions than the detected regions by the NVT algorithm. The difference is notorious when the five most relevant regions were counted. This fact could be useful when we want to lead an autonomous robotic exploration to gather video-observations of this kind of environments (coral reefs), because if more regions are detected in the coral reef then the autonomous agent will go to that place instead of moving to a zone where there is only water.

TABLE 1: Comparison of the five most relevant regions detected as relevant or interesting (nonwater regions) by using the NVT and the proposed method. In the last row the total number of images and the average of the percentage of detected relevant regions are specified.

Seq.	Frames	Depth [m]	% of interesting regions (NVT)	% of interesting regions (AVA)
1	168	7.7	75.20	95.85
2	243	7.8	66.91	95.80
3	153	7.8	83.00	99.60
4	181	11.8	57.79	92.15
5	163	7.1	74.47	98.28
6	242	11.3	59.92	90.76
1150			<b>69.04</b>	<b>94.90</b>

**4.3. Tracking of Relevant Regions.** In this section, a comparison between the tracking of a region by using the superpixel descriptors and a keypoint-based descriptor is done. As keypoint detector and descriptor we have used SURF [15], SIFT [16], and ORB [32]. The implementation of the descriptors is the one available on OpenCV. To find the correspondence between keypoints we have use a robust matcher which is available in [33]. The keypoint descriptors and detectors are used with default configuration. For AVA, the yellow and red features have preference through the weights.

For this test, we evaluate the length of tracking, that is, the number of consecutive frames that a given region is tracked in a sequence of images. The region to be tracked is the most relevant as considered by the proposed visual

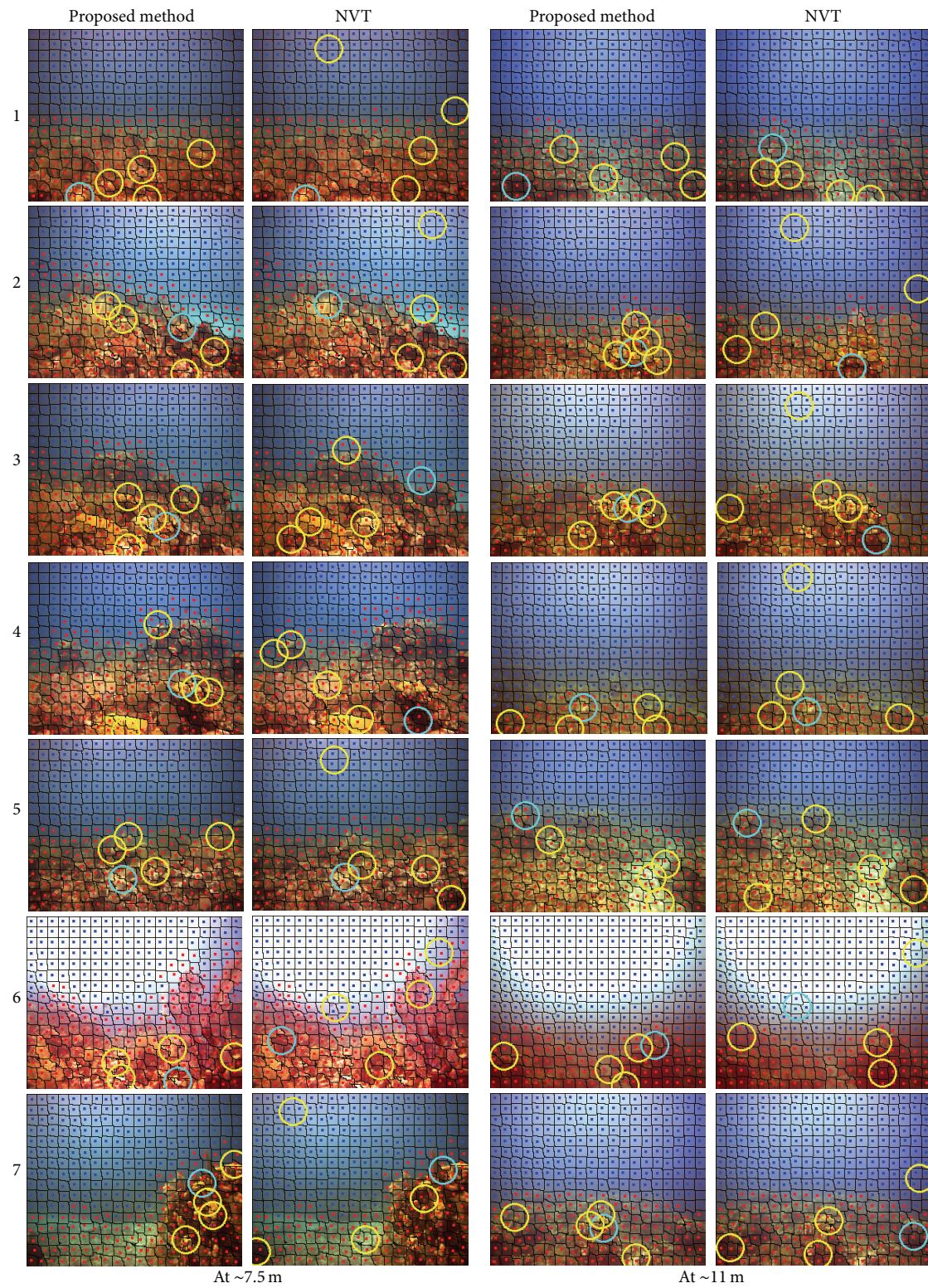


FIGURE 12: Some of regions detected as interesting by our visual attention algorithm and the NVT. The videos sequences were taken approximately between 7.5 m and 11 m of depth.



TABLE 2: Comparison of the most relevant region detected as relevant (nonwater regions) by using the NVT and the proposed method. In the last row the total number of images and the average of the percentage of detected relevant regions are specified.

Seq.	Frames	Depth [m]	% of <i>relevant</i> region (NVT)	% of <i>relevant</i> region (AVA)
1	168	7.7	90.47	91.07
2	243	7.8	94.23	95.06
3	153	7.8	99.34	100
4	181	11.8	88.39	95.5
5	163	7.1	95.02	98.15
6	242	11.3	87.02	91.60
1150			<b>92.41</b>	<b>95.23</b>

TABLE 3: Tracking length comparison of SURF, SIFT, and ORB against the proposed approach.

Tracker	Processing time per frame	Tracking length
SURF	227.8 ms	64.67%
SIFT	258.6 ms	74.35%
ORB	13.6 ms	32.65%

attention algorithm. The image sequences are taken from different videos recorded by a diver while exploring a coral reef.

It is important to remark that the complexity of the AVA algorithm is  $O(N)$ , where  $N$  is the total number of pixels in the image. The average processing time, in a 2.1 GHz dual-core processor, for an image of  $480 \times 270$  is 122 ms. A total of 8545 images comprises the sequences used in this test.

In Table 3, the average percentages of length of tracking between the proposed method and the keypoint-based trackers and the average processing times are shown. The percentage indicates how a long keypoint-based method's tracking length is in comparison with the AVA tracking length. For example, the tracking length of the SIFT-based tracker is 74.3% of the AVA tracking length. We have normalized all the percentages with respect to the AVA tracking length because it was the method that gets the longer tracking length.

Although the SIFT-based tracker is the one with almost the same tracking length as AVA, it is approximately twice slower. The faster tracker is the one based on ORB; however, it is also the one with the smallest tracking length. From the results of Table 3, it can be noted that the proposed approach outperforms the other methods when tracking regions in underwater environments, particularly coral reefs.

With respect to the processing time, our method can process in average 8 frames of size  $480 \times 270$  per second. It is important to consider that the current implementation of our

method is not yet optimized in terms of software. However, we have found that the current processing frame rate can be good enough to work when exploring an underwater environment because this task tends to be executed with slow motions of the AUV.

From the presented results, it can be remarked that the proposed method can detect and track regions that are likely to draw the attention of humans in coral reefs. This makes our approach suitable for using it to guide an exploration in terms of regions of potential interest for humans.

**4.4. Field Trials.** For experimental tests we use an amphibious robot named Mexibot of the AQUA family [34]. In water, the robot's propulsion is based on six fins that can provide motion in 5 degrees of freedom up to depths near 35 meters. Mexibot's medium size ( $60 \times 45 \times 12$  cm) allows for easy maneuverability, which is important in the time response on the robot's control, when navigating with the purpose of closely monitoring an unstructured environment.

All the trials were performed in an area that belongs to the second largest coral reef system, located in Costa Maya, Mexico. The coral reef ecosystem in this zone has a wide diversity of living organisms (flora and fauna) with a great variety in colors. It also has variable conditions in terms of depth and visibility. We performed the experiments in a depth range from 5 to 18 m.

During the field trials several exploration tests were performed. Most of the tests were set to a two-minute duration as we needed to verify their performance under different conditions. In Figure 13, the results from an exploration are shown. During this test the AUV was programmed to turn  $90^\circ$  around its  $Z$  axis every certain time. This had two purposes: the first one is for safety, to avoid a possible collision between the robot and the coral reef. In the moment of the tests the AUV did not have an implemented method for collision avoidance. The second purpose is for testing the capabilities of the proposed approach to detect and track new regions. This way the AUV should detect and track a different region every certain time.

It can be seen in Figure 13 that the AUV effectively changes its yaw angle in order to track the region detected by the visual attention algorithm. The boxes in Figures 13(a) and 13(b) enclose the period during which the same RoI was tracked by the AUV. It can be seen that these regions were followed during several seconds by the AUV until before the  $90^\circ$  turn. These results show that the proposed approach can be used to guide the motion of a AUV for exploring an unknown environment.

## 5. Conclusions and Future Work

We have presented ongoing research on the detection and tracking of invariant features that are considered relevant during the exploration of a coral reef habitat. The main goal is to perform an autonomous cautious exploration and gather high quality image data with a robotic system that could be directly deployed into the environment, with few or no prior information of it. It is important to highlight that

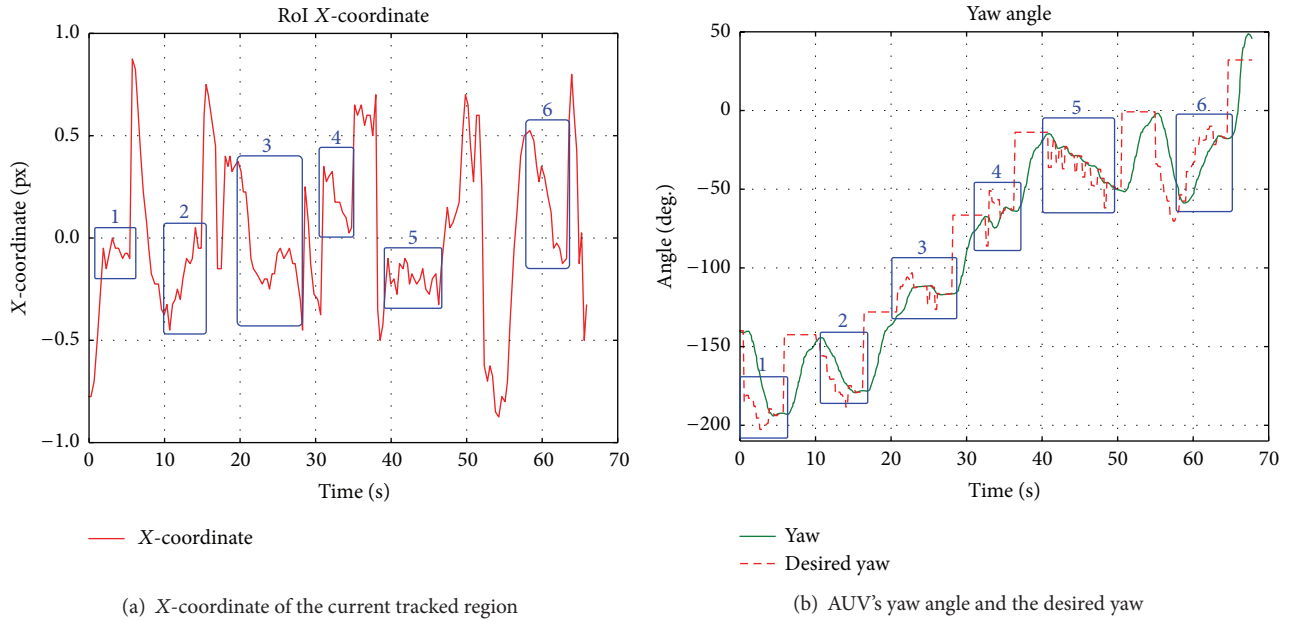


FIGURE 13: Obtained results during a field trial in a coral reef. In (a) and (b) we can observe the X-coordinate of the region of interest as well as the yaw angle of the AUV, respectively. In (c) images from the tracked RoI can be seen.

the system is trained to adapt itself to the local water and illumination conditions in an online manner. The integrated framework is fast enough to perform the exploration while fitting to the control navigation requirements of the system. Future research will focus on the incorporation of a notion of forward movement to estimate how far the robot is from a certain region as well as adding texture information on the

detection of regions of interest in order to reduce errors in the selection of relevant regions (e.g., sand or rock regions are not of interest for exploration).

### Competing Interests

The authors declare that they have no competing interests.



## Acknowledgments

The authors would like to thank CONACyT for their support and project funding. The authors also thank Mar Adentro Diving, Mahahual, for their support during their sea trials.

## References

- [1] Y. Girdhar, P. Giguère, and G. Dudek, "Autonomous adaptive exploration using realtime online spatiotemporal topic modeling," *International Journal of Robotics Research*, vol. 33, no. 4, pp. 645–657, 2014.
- [2] Y. Girdhar and G. Dudek, "Exploring underwater environments with curiosity," in *Proceedings of the 11th Conference on Computer and Robot Vision (CRV '14)*, pp. 104–110, IEEE, Montreal, Canada, May 2014.
- [3] A. Maldonado-Ramírez and L. A. Torres-Méndez, "Using super-color pixels descriptors for tracking relevant cues in underwater environments with poor visibility conditions," in *Proceedings of the IEEE Workshop on Visual Place Recognition in Changing Environments (ICRA '15)*, May 2015.
- [4] A. Maldonado-Ramírez, L. Torres-Méndez, and E. Martínez-García, "Robust detection and tracking of regions of interest for autonomous underwater robotic exploration," in *Proceedings of the 6th International Conference on Advanced Cognitive Technologies and Applications*, pp. 165–171, Venice, Italy, May 2014.
- [5] Y. Y. Schechner and N. Karpel, "Clear underwater vision," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 1, pp. 1536–1543, IEEE, Washington, DC, USA, June 2004.
- [6] CIE, *Recommendations on Uniform Color Spaces, Color Difference Equations, Psychometric Color Terms*, vol. 2, no. 15 (E.-1.3.1), CIE Publication, 1971.
- [7] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, 2001.
- [8] L. F. M. Vieira, E. R. D. Nascimento, F. A. Fernandes Jr., R. L. Carceroni, R. D. Vilela, and A. D. A. Araújo, "Fully automatic coloring of grayscale images," *Image and Vision Computing*, vol. 25, no. 1, pp. 50–60, 2007.
- [9] G. Bianco, M. Muzzupappa, F. Bruno, R. Garcia, and L. Neumann, "A new color correction method for underwater imaging," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 5, pp. 25–32, 2015.
- [10] W. James, *The Principles of Psychology*, 1890.
- [11] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," in *Matters of Intelligence: Conceptual Structures in Cognitive Neuroscience*, L. Vaina, Ed., vol. 188 of *Synthese Library: Studies in Epistemology, Logic, Methodology, and Philosophy of Science*, pp. 115–141, Springer, Amsterdam, The Netherlands, 1987.
- [12] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [13] S. Frintrop, E. Rome, and H. I. Christensen, "Computational visual attention systems and their cognitive foundations: a survey," *ACM Transactions on Applied Perception*, vol. 7, no. 1, article 6, 2010.
- [14] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference*, vol. 15, p. 50, Manchester, UK, 1988.
- [15] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: speeded up robust features," in *Computer Vision-ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds., vol. 3951 of *Lecture Notes in Computer Science*, pp. 404–417, Springer, Berlin, Germany, 2006.
- [16] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, IEEE, 1999.
- [17] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.
- [18] C. Akinlar and C. Tonal, "EDCircles: real-time circle detection by Edge Drawing (ED)," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '12)*, pp. 1309–1312, Kyoto, Japan, March 2012.
- [19] S. Frintrop, G. Backer, and E. Rome, "Goal-directed search with a top-down modulated computational attention system," in *Pattern Recognition*, W. Kropatsch, R. Sablatnig, and A. Hanbury, Eds., vol. 3663 of *Lecture Notes in Computer Science*, pp. 117–124, Springer, Berlin, Germany, 2005.
- [20] M. Begum and F. Karray, "Visual attention for robotic cognition: a survey," *IEEE Transactions on Autonomous Mental Development*, vol. 3, no. 1, pp. 92–105, 2011.
- [21] D. Walther, D. R. Edgington, and C. Koch, "Detection and tracking of objects in underwater video," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 1, pp. I-544–I-549, July 2004.
- [22] D. Edgington, K. Salamy, M. Risi, R. E. Sherlock, D. Walther, and C. Koch, "Automated event detection in underwater video," in *Proceedings of the in OCEANS 2003*, vol. 5, pp. P2749–P2753, San Diego, Calif, USA, September 2003.
- [23] C. Barat and M.-J. Rendas, "A robust visual attention system for detecting manufactured objects in underwater video," in *Proceedings of the OCEANS*, pp. 1–6, Singapore, May 2006.
- [24] P. L. Correia, P. Y. Lau, P. Fonseca, and A. Campos, "Underwater video analysis for norway lobster stock quantification using multiple visual attention features," in *Proceedings of the 15th European Signal Processing Conference*, pp. 1764–1768, IEEE, Poznan, Poland, 2007.
- [25] S. Frintrop, *Vocus: a visual attention system for object detection and goal-directed search [Ph.D. dissertation]*, Rheinische Friedrich-Wilhelms-Universität, Bonn, Germany, 2006.
- [26] S. Palmer, *Vision Science, Photons to Phenomenology*, The MIT Press, 1999.
- [27] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994.
- [28] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels," Tech. Rep., EPFL, Lausanne, Switzerland, 2010.
- [29] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2281, 2012.
- [30] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, no. 9, pp. 1395–1407, 2006.

- [31] F. G. Rodríguez-Telles, L. A. Torres-Méndez, and E. A. Martínez-García, "A fast floor segmentation algorithm for visual-based robot navigation," in *Proceedings of the 10th International Canadian Conference on Computer and Robot Vision (CRV '13)*, pp. 167–173, May 2013.
- [32] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proceedings of the International Conference on Computer Vision (ICCV '11)*, pp. 2564–2571, IEEE, Barcelona, Spain, November 2011.
- [33] R. Laganière, *OpenCV 2 Computer Vision Application Programming Cookbook: Over 50 Recipes to Master This Library of Programming Functions for Real-Time Computer Vision*, Packt Publishing, 2011.
- [34] G. Dudek, P. Giguere, J. Zacher et al., "Aqua: an amphibious autonomous robot," *Computer*, vol. 40, no. 1, pp. 46–53, 2007.

## Research Article

# Vision-Based Autonomous Underwater Vehicle Navigation in Poor Visibility Conditions Using a Model-Free Robust Control

Ricardo Pérez-Alcocer,<sup>1</sup> L. Abril Torres-Méndez,<sup>2</sup>  
Ernesto Olguín-Díaz,<sup>2</sup> and A. Alejandro Maldonado-Ramírez<sup>2</sup>

<sup>1</sup>CONACYT-Instituto Politécnico Nacional-CITEDI, 22435 Tijuana, BC, Mexico

<sup>2</sup>Robotics and Advanced Manufacturing Group, CINVESTAV Campus Saltillo, 25900 Ramos Arizpe, COAH, Mexico

Correspondence should be addressed to L. Abril Torres-Méndez; [abriltorresm15@gmail.com](mailto:abriltorresm15@gmail.com)

Received 25 March 2016; Revised 5 June 2016; Accepted 6 June 2016

Academic Editor: Pablo Gil

Copyright © 2016 Ricardo Pérez-Alcocer 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 vision-based navigation system for an autonomous underwater vehicle in semistructured environments with poor visibility. In terrestrial and aerial applications, the use of visual systems mounted in robotic platforms as a control sensor feedback is commonplace. However, robotic vision-based tasks for underwater applications are still not widely considered as the images captured in this type of environments tend to be blurred and/or color depleted. To tackle this problem, we have adapted the  $l\alpha\beta$  color space to identify features of interest in underwater images even in extreme visibility conditions. To guarantee the stability of the vehicle at all times, a model-free robust control is used. We have validated the performance of our visual navigation system in real environments showing the feasibility of our approach.

## 1. Introduction

The development of research in autonomous underwater vehicles (AUVs) began approximately four decades ago. Since then, a considerable amount of research has been presented. In particular, the localization and navigation problems represent a challenge in the AUVs development due to the unstructured and hazardous conditions of the environment and the complexity of determining the global position of the vehicle. An extensive review of the research related to this topic is presented in [1–4].

Sensor systems play a relevant role in the development of AUV navigation systems as they provide information about the system status and/or environmental conditions. There exist several sensors that provide relevant and accurate information [5–7]. However, global or local pose estimation of underwater vehicles is still an open problem, specially when a single sensor is used. Typically, underwater vehicles use multisensor systems with the intention of estimating their position and determining the location of objects in their workspace. Usually, inertial measurement units (IMUs),

pressure sensors, compasses, and global positioning systems (GPS) are commonly used [8]. Note that even though GPS devices are widely used for localization, they show low performance in an underwater environment. Therefore, data fusion is needed to increase the accuracy of the pose estimation (for a review of sensor fusion techniques see [9].)

Vision-based systems are a good choice because they provide high resolution images with high speed acquisition at low cost [10]. However, in aquatic environments the color attenuation produces poor visibility when the distance increases. In contrast, at short distances the visibility may be good enough and the measurement accuracy higher than other sensors. Therefore, tasks in which visual information is used are limited to object recognition and manipulation, docking vehicle [11], reconstruction of the ocean floor structure [12], and underwater inspection and maintenance [13]. In [14], the authors discuss how visual systems can be used in underwater vehicles, and they present a vision system which obtains depth estimations based on a camera data. In [10], a visual system was introduced. This visual system, called Fugu-f, was designed to provide visual information in

submarine tasks such as navigation, surveying, and mapping. The system is robust in mechanical structure and software components. Localization has been also addressed with vision systems. In [15] a vision-based localization system for an AUV with limited sensing and computation capabilities was presented. The vehicle pose is estimated using an Extended Kalman Filter (EKF) and a visual odometer. The work in [16] presents a vision-based underwater localization technique in a structured underwater environment. Artificial landmarks are placed in the environment and a visual system is used to identify the known objects. Additionally a Monte Carlo localization algorithm estimates the vehicle position.

Several works for visual feedback control in underwater vehicles have been developed [17–28]. In [17], the authors present a Boosting algorithm which was used to identify features based on color. This method uses, as input, images in the RGB color space, and a set of classifiers are trained offline in order to segment the target object to the background and the visual error is defined as an input signal for the PID controller. In a similar way, a color-based classification algorithm is presented in [18]. This classifier was implemented using the JBoost software package in order to identify buoys of different color. Both methods require an offline training process which is a disadvantage when the environment changes. In [19], an adaptive neural network image-based visual servo controller is proposed; this control scheme allows placing the underwater vehicle in the desired position with respect to a fixed target. In [20], a self-triggered position based visual servo scheme for the motion control of an underwater vehicle was presented. The visual controller is used to keep the target in the center of image with the premise that the target will always remain inside the camera field of view. In [21], the authors present an evolved stereo-SLAM procedure implemented in two underwater vehicles. They computed the pose of the vehicle using a stereo visual system and the navigation was performed following a dynamic graph. A visual guidance and control methodology for a docking task is presented in [22]. Only one high-power LED light was used for AUV visual guidance without distance estimation. The visual information and a PID controller were employed in order to regulate the AUV attitude. In [23], a robust visual controller for an underwater vehicle is presented. The authors implemented genetic algorithms in a stereo visual system for real-time pose estimation, which was tested in environments under air bubble disturbance. In [24], the development and testing process of a visual system for buoys detection is presented. This system used the HSV color space and the Hough transformation in the detection process. These algorithms require the internal parameters adjusting depending on the work environment, which is a disadvantage. In general, the visual systems used in these papers were configured for a particular environment and when the environmental characteristics change, it is necessary to readjust some parameters. In addition, robust control schemes were not proposed for attitude regulation.

In this work, a novel navigation system for autonomous underwater vehicle is presented. The navigation system combines a visual controller with an inertial controller in

order to define the AUV behavior in a semistructured environment. The AUV dynamic model is described and a robust control scheme is experimentally validated for attitude and depth regulation tasks. An important controller feature is that it can be easily implemented in the experimental platform. The main characteristics affecting the images taken underwater are described, and an adapted version of the perceptually uniform color space  $l\alpha\beta$  is used to find the artificial marks in a poor visibility environment. The exact positions of the landmarks in the vehicle workspace are not known, but an approximate knowledge of their localization is available.

The main contributions of this work include (1) the development of a novel visual system for detection of artificial landmarks in poor visibility conditions underwater, which does not require the adjustment of internal parameters when environmental conditions change, and a new simple visual navigation approach which does not require keeping the objects of interest in the field of view of the camera at all times, considering that only their approximate localization is given. In addition, a robust controller guarantees the stability of the AUV.

The remaining part of this paper is organized as follows. In Section 2 the visual system is introduced. The visual navigation system and details of the controller are presented in Section 3. Implementation details and the validated experimental results are presented in Section 4. Finally, Section 5 concludes this work.

## 2. The Visual System

Underwater visibility is poor due to the optical properties of light propagation, namely, absorption and scattering, which are involved in the image formation process. Although a big amount of research has focused on using mathematical models for image enhancement and restoration [25, 26], it is clear that the main challenge is the highly dynamic environment; that is, the limited number of parameters that are typically considered could not represent all the actual variables involved in the process. Furthermore, for effective robot navigation, the enhanced images are needed in real time, which is not always possible to achieve in all approaches. For that reason, we decided to explore directly the use of perceptually uniform color spaces, in particular the  $l\alpha\beta$  color space. In the following sections, we describe the integrated visual framework proposed for detecting artificial marks in aquatic environments, in which the  $l\alpha\beta$  color space was adapted for underwater imagery.

*2.1. Color Discrimination for Underwater Images Using the  $l\alpha\beta$  Color Space.* Three main problems are observed in underwater image formation [26]. The first is known as disturbing noise, which is due to suspended matter in water, such as bubbles, small particles of sand, and small fish or plants that inhabit the aquatic ecosystem. These particles block light and generate noisy images with distorted colors. The second problem is related with the refraction of light. When a camera set and objects are placed in two different





FIGURE 1: Photographs with multicolored objects taken underwater and in air.

environments with different refractive index, the objects in the picture have different distortion for each environment, and therefore the position estimation is not the same in both environments. The third problem in underwater images is the light attenuation. The light intensity decreases as the distance to the objects increases; this is due to the attenuation of the light in function of its wavelength. The effect of this is that the colors of the observed underwater objects look different from those perceived in the air. Figure 1 shows two images with the same set of different color objects taken underwater and in air. In these images, it is possible to see the characteristics of the underwater images mentioned above.

A color space is a mathematical model through which the perceptions of color are represented. The color space selection is an important decision in the development of the image processing algorithm, because it can dramatically affect the performance of the vision system. We selected the  $l\alpha\beta$  color space [27], because it has features that simplify the analysis of the data coming from the underwater images. In underwater images, the background color (sea color) is usually blue or green; these colors correspond to the limits of the  $\alpha$  and  $\beta$  channels, respectively, and, therefore, to identify objects with contrasting colors to the blue or green colors results much easier. A modification of the original transformation method from the RGB to the  $l\alpha\beta$  space color was made. The logarithm operation was removed from the transformation reducing the processing time while keeping the color distribution. Thus, the mapping between RGB and the modified  $l\alpha\beta$  color space is expressed as a linear transformation:

$$\begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0.3475 & 0.8265 & 0.5559 \\ 0.2162 & 0.4267 & -0.6411 \\ 0.1304 & -0.1033 & -0.0269 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \quad (1)$$

where  $l \in [0.0, 1.73]$  is the achromatic channel which determines the luminance value,  $\alpha \in [-0.6411, 0.6429]$  is the yellow-blue opposite channel, and  $\beta \in [-0.1304, 0.1304]$ , is the red-cyan with a significant influence of green. The data in these channels include a wide variety of colors; however, the information in aquatic images is contained in a very narrow interval. Figure 2 shows an underwater image and the frequency histogram for each channel of the  $l\alpha\beta$  color space.

In this image, the data of the objects are concentrated in a small interval.

Therefore, in order to increase the robustness of the identification method, new limits for each channel are established. These values help to increase the contrast between objects and the background in the image. The new limits are calculated using the frequency histogram for each of the channels, and, with this, the extreme values in the histogram with a higher frequency than a threshold value are computed. The difference between using the frequency histogram, and not only the minimum and maximum values, is that the first method eliminates outliers.

Finally, a data normalization procedure is performed using the new interval in each channel of the  $l\alpha\beta$  color space. After this, it is possible to obtain a clear segmentation of the objects with colors located at the end values of the channels. Figure 3 shows the result of applying the proposed algorithm in the  $l$ ,  $\alpha$ ,  $\beta$  channels. It can be observed that some objects are significantly highlighted from the greenish background; particularly, the red circle in the beta channel presents a high contrast.

## 2.2. Detection of Artificial Marks in Aquatic Environments.

The localization problem for underwater vehicles requires identifying specific objects in the environment. Our navigation system relies on a robust detection of the artificial marks in the environment. Artificial red balls were selected as the known marks in the aquatic environment. Moreover, circles tags with different color were attached to the sphere in order to determine the section on the sphere that is being observed.

Detection of circles in images is an important and frequent problem in image processing and computer vision. A wide variety of applications such as quality control, classification of manufactured products, and iris recognition use circle detection algorithms. The most popular techniques for detecting circles are based on the Circle Hough Transform (CHT) [28]. However, this method is slow, demands a considerable amount of memory, and identifies many false positives, especially in the presence of noise. Furthermore, it has many parameters that must be previously selected by the user. This last characteristic limits their use in underwater environments since ambient conditions are constantly changing. For this reason, it is desirable a circle detection

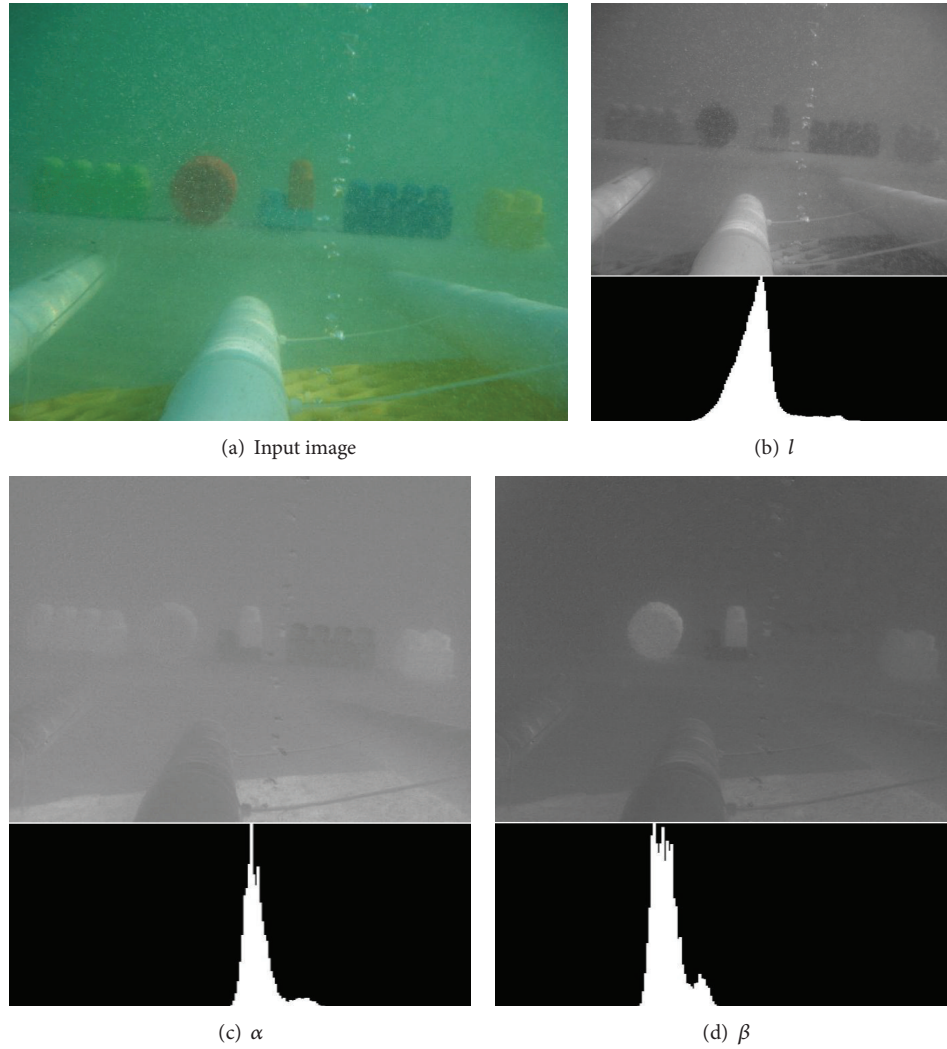


FIGURE 2: Frequency histograms for each channel of  $l\alpha\beta$  color space.

algorithm with a fixed set of internal parameters that does not require adjustment even if small or large circle identification is required or if the ambient light changes. The circle detection algorithm presented by Akinlar and Tobal [29] provides the desired properties. We have evaluated its performance in aquatic images with good results. Specifically, we applied the algorithm to the  $\beta$  channel image which is the resulting image from the procedure described in the previous section. As it was mentioned, the  $\beta$  channel presents the highest contrast between red color objects and the background color of underwater images. This enables the detection algorithm to find circular shapes in the field of view with more precision. This is an important discover to the best of our knowledge, this is the first time that this color space model is used in underwater images for this purpose.

Figure 4 shows the obtained results. The images are organized as follows: the first column shows the original input image; the second column corresponds to the graphical representation of the  $\beta$  channel; and finally the third column

displays the circles detected in the original image. The rows in the figure present the obtained results under different conditions. The first experiment analyzes a picture taken in a pool with clear water. Although the spheres are not close to the camera, they can be easily detected by our visual system. The second row is also a photograph taken in the pool, but in this case the visibility was poor; however, the method works appropriately and detects the circle. Finally, the last row shows the results obtained from a scene taken in the marine environment, in which visibility is poor. In this case, the presence of the red object in the image is almost imperceptible to the human eye; however the detector identifies the circle successfully.

The navigation system proposed in this work is the integration of the visual system, described above, with a novel control scheme that defines the behavior of the vehicle based on the available visual information. The block diagram in Figure 5 shows the components of the navigation system and the interaction between them.

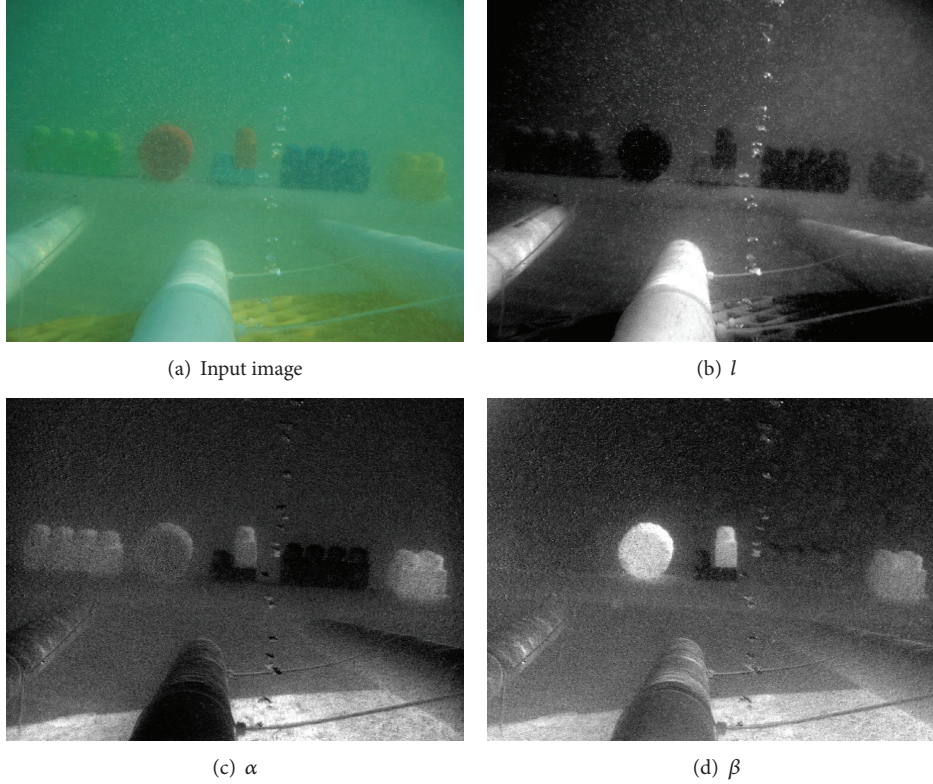


FIGURE 3: Result of conversion of the input image to color space  $l\alpha\beta$  after adjusting the range of values.

### 3. Visual Navigation System

In this section the navigation system and its implementation in our robotic system are presented. The autonomous underwater vehicle, called Mexibot (see Figure 6), is part of the Aqua robot family [30] and an evolution of the RHex platform [31]. The Aqua robots are amphibious with the ability to work in both land and water environments. The underwater vehicle has a pair of embedded computers; one computer is used for the visual system and for other phases such as registration of data; the second computer is used for the low-level control. An important feature is that both computers are connected via Ethernet, so they are able to exchange data or instructions. The control loop of the robot runs on a real-time constraint; for this reason, QNX operating system is installed in the control computer. On the other hand, the vision computer has Ubuntu 12.04 as the operating system. On this computer, high-level applications are developed which use the Robot Operating System (ROS). In addition, the vehicle has an IMU, which provides attitude and angular velocity of the vehicle. A pressure sensor is used to estimate the depth of the robot, and a set of three cameras are used, two in front of the robot and one in the back.

**3.1. Model-Free Robust Control.** The visual navigation system requires a control scheme to regulate the depth and attitude of the underwater vehicle. In this subsection, the underwater vehicle dynamics is analyzed and the controller used

to achieve the navigation objective is presented. Knowing the dynamics of underwater vehicles and their interaction with the environment plays a vital role for the vehicles performance. The underwater vehicles dynamics include hydrodynamic parametric uncertainties, which are highly nonlinear, coupled, and time varying. The AUV is a rigid body moving in 3D space. Consequently, the AUV dynamics can be represented with respect to the inertial reference denoted by  $I = \{e_x \ e_y \ e_z\}$  or with respect to the body reference frame  $B = \{e_x^b \ e_y^b \ e_z^b\}$ . Figure 7 presents the AUV reference frames and their movements.

In [32], Fossen describes the method to obtain the underwater vehicle dynamics using Kirchhoff's laws. Fluid damping, gravity-buoyancy, and all external forces are also included and the following representation is obtained:

$$M\dot{\mathbf{v}} + C_v(\mathbf{v})\mathbf{v} + D_v(\mathbf{v}_R)\mathbf{v} + \mathbf{g}_v(\mathbf{q}) = \mathbf{F}_u + \boldsymbol{\eta}_v(\cdot) \quad (2)$$

$$\mathbf{v} = J(\mathbf{q})\dot{\mathbf{q}}, \quad (3)$$

where  $\mathbf{q} = [x \ y \ z \ \phi \ \theta \ \psi]^T \in \mathbb{R}^6$  is the pose of the vehicle,  $\mathbf{v} = [\mathbf{v}^T, \boldsymbol{\omega}^T]^T \in \mathbb{R}^6$  is the twist of the vehicle,  $\mathbf{v} = [v_x \ v_y \ v_z]^T \in \mathbb{R}^3$  is the linear velocity, and  $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]^T \in \mathbb{R}^3$  is the angular velocity expressed in the body reference frame.  $M \in \mathbb{R}^{6 \times 6}$  is the positive constant and symmetric inertia matrix which includes the inertial mass



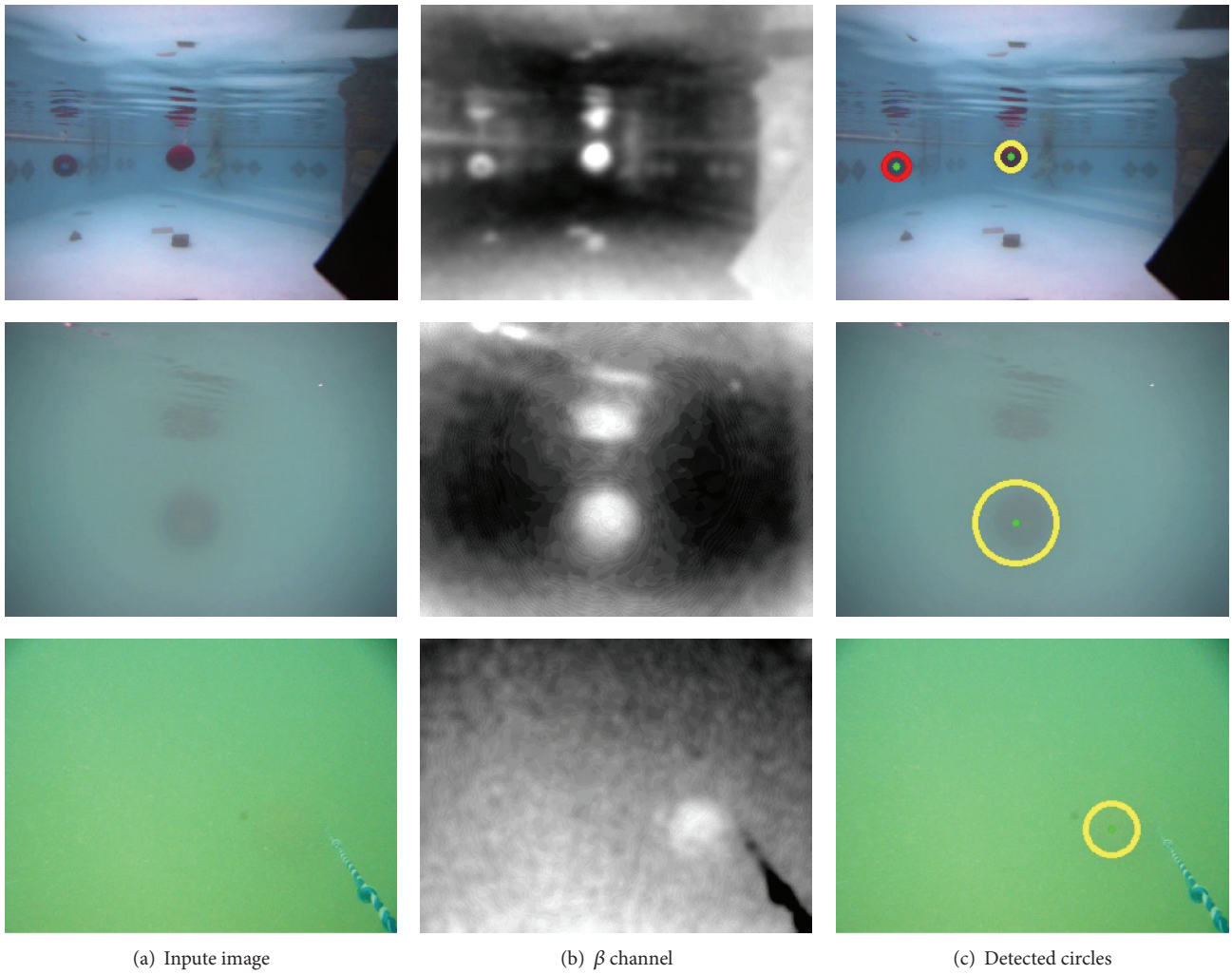


FIGURE 4: Example results of applying the circle detection algorithm using the  $l\alpha\beta$  color space in underwater images with different visibility conditions.

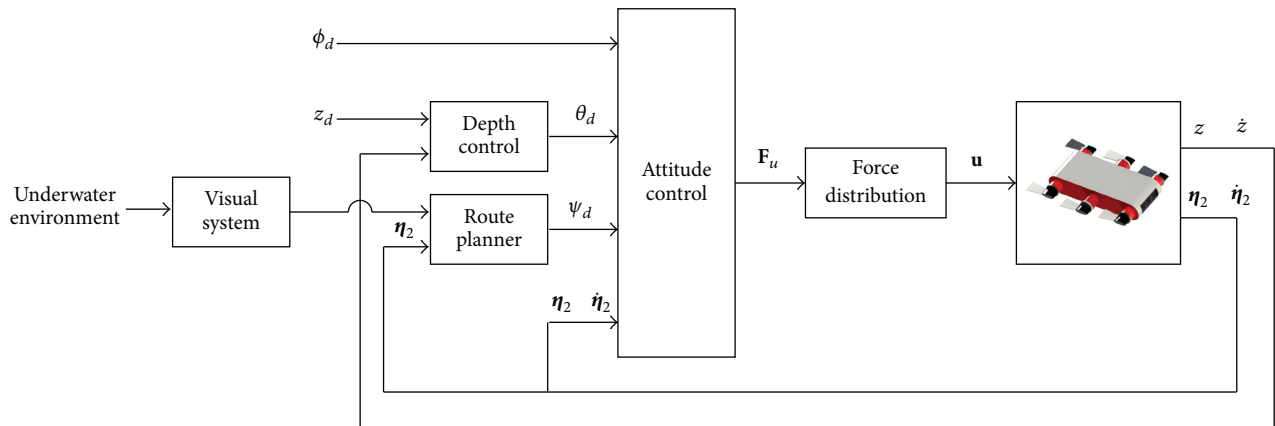


FIGURE 5: Block diagram of the proposed navigation system for autonomous underwater vehicle.





FIGURE 6: Our underwater vehicle Mexibot.

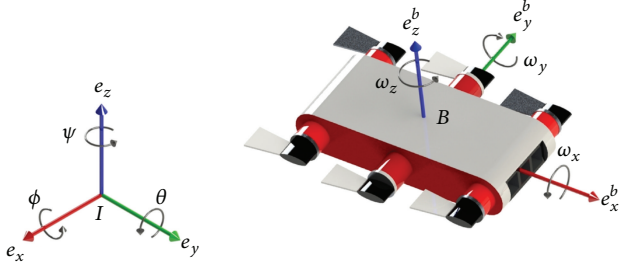


FIGURE 7: AUV representation including the inertial and body reference frame.

and the added mass matrix.  $C_v(\mathbf{v}) \in \mathbb{R}^{6 \times 6}$  is the skew-symmetric Coriolis matrix and  $D_v(\mathbf{v}_R) > 0 \in \mathbb{R}^{6 \times 6}$  is the positive definite dissipative matrix, which depends on the magnitude of the relative fluid velocity  $\mathbf{v}_R \in \mathbb{R}^6$ .  $\boldsymbol{\zeta} \in \mathbb{R}^6$  is the fluid velocity in the inertial reference and  $\mathbf{g}_v(\mathbf{q}) \in \mathbb{R}^6$  is the potential wrench vector which includes gravitational and buoyancy effects.  $\mathbf{F}_u \in \mathbb{R}^6$  is the vector of external forces, expressed in the vehicle frame and produced by the vehicle thrusters,  $J(\mathbf{q}) \in \mathbb{R}^{6 \times 6}$  is the operator that maps the generalized velocity  $\dot{\mathbf{q}}$  to the vehicle twist  $\mathbf{v}$ , and  $\boldsymbol{\eta}(\cdot)$  is the external disturbance wrench produced by the fluid currents.

Consider the following control law [33]:

$$\mathbf{F}_u = \widehat{M}\dot{\mathbf{v}}_r + \widehat{D}_v\mathbf{v}_r - J^{-T}(\mathbf{q}) \left( K_s \mathbf{s} + K_i \int \mathbf{s} dt + \beta \|\mathbf{s}\|^2 \mathbf{s} \right), \quad (4)$$

where  $\widehat{M}$ ,  $\widehat{D}_v$ ,  $K_s$ ,  $K_i$ , and  $\Lambda$  are constant positive definite matrices,  $\beta$  is a positive scalar, and  $\tilde{\mathbf{q}}$  is the pose error:

$$\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d, \quad (5)$$

after which the extended (tracking) error  $\mathbf{s}$  is defined as

$$\mathbf{s} = \dot{\tilde{\mathbf{q}}} + \Lambda \tilde{\mathbf{q}}. \quad (6)$$

Expressing this extended error as a velocity error

$$\mathbf{s} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r \quad (7)$$

for an artificial reference velocity  $\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \Lambda \tilde{\mathbf{q}}$ , it raises the vehicle's twist reference as

$$\mathbf{v}_r \triangleq J(\mathbf{q})\dot{\mathbf{q}}_r = J(\mathbf{q})(\dot{\mathbf{q}}_d - \Lambda \tilde{\mathbf{q}}) = \mathbf{v}_d - J(\mathbf{q})\Lambda \tilde{\mathbf{q}}. \quad (8)$$

This control scheme ensures stability for tracking tasks despite any inaccuracies in the dynamic parameters of the vehicle and the perturbations in the environment, [33]. Therefore, this control law can be used to define the behavior of both the inertial and the visual servoing mode of the underwater vehicle.

It is also important to highlight that this control law can be implemented easily, because it only requires measurements of  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$  and rough estimates of  $M$  and  $D_v$ .

**3.1.1. Stability Analysis.** Model (2)-(3) is also known as the quasi-Lagrangian formulation since the velocity vector  $\mathbf{v}$  defines a quasi-Lagrangian velocity vector. The Lagrangian formulation upon which the stability analysis relies is found by using (3) and its time derivative on (2) and premultiply the resulting equation by the transpose of the velocity operator  $J^T(\mathbf{q})$  [34]:

$$H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + D(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v}_R)\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\eta}(\cdot), \quad (9)$$

where  $H(\mathbf{q}) = J^T(\mathbf{q})MJ(\mathbf{q}) = H^T(\mathbf{q}) > 0$ ;  $C(\mathbf{q}, \dot{\mathbf{q}}) = J^T(\mathbf{q})M\dot{J}(\mathbf{q}) + J^T(\mathbf{q})C_vJ(\mathbf{q})$ , which implies that  $C - (1/2)\dot{H} = Q$ ;  $Q + Q^T = 0$ ; and all the terms are bounded, for nonnegative constants  $b_i \geq 0$  as follows:

$$\begin{aligned} \|H(\mathbf{q})\| &\leq b_1 = \lambda_M\{H(\mathbf{q})\} \\ \|C(\mathbf{q}, \dot{\mathbf{q}})\| &\leq b_2 \|\dot{\mathbf{q}}\| \\ \|D(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v}_R)\| &\leq b_3 \|\dot{\mathbf{q}}\| + b_4 \|\boldsymbol{\zeta}\| \\ \|\mathbf{g}(\mathbf{q})\| &\leq b_5 \\ \|\boldsymbol{\eta}\| &\leq b_6 \|\boldsymbol{\zeta}\| + b_7 \|\dot{\mathbf{q}}\| \|\boldsymbol{\zeta}\| + b_8 \|\boldsymbol{\zeta}\|^2. \end{aligned} \quad (10)$$

Then, control law (4) adopts the following shape in the Lagrangian space:

$$\begin{aligned} \boldsymbol{\tau} &= J^T(\mathbf{q})\mathbf{F}_u \\ &= \widehat{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \widehat{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \widehat{D}(\mathbf{q})\dot{\mathbf{q}}_r - K_s \mathbf{s} \\ &\quad - K_i \int \mathbf{s} dt - \beta \|\mathbf{s}\|^2 \mathbf{s}, \end{aligned} \quad (11)$$

with the following relationships:

$$\begin{aligned} \widehat{H}(\mathbf{q}) &\triangleq J^T(\mathbf{q})\widehat{M}J(\mathbf{q}) > 0, \\ \widehat{C}(\mathbf{q}, \dot{\mathbf{q}}) &\triangleq J^T(\mathbf{q})\widehat{M}\dot{J}(\mathbf{q}), \\ \widehat{D}(\mathbf{q}) &\triangleq J^T(\mathbf{q})\widehat{D}_vJ(\mathbf{q}) > 0, \end{aligned} \quad (12)$$

from which it raises  $\widehat{C} + \widehat{C}^T = \dot{\widehat{H}}$  or equivalently the following property:

$$\mathbf{x}^T \left[ \frac{1}{2} \dot{\widehat{H}}(\mathbf{q}) - \widehat{C}(\mathbf{q}, \dot{\mathbf{q}}) \right] \mathbf{x} = 0, \quad \forall \mathbf{x} \neq 0. \quad (13)$$

Now consider that the left-hand side of Lagrangian formulation (9) can be expressed in the following regression-like expression:

$$H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + D(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{r}_R)\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = Y(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\theta}, \quad (14)$$

where  $Y(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) : \mathbb{R}^P \mapsto \mathbb{R}^n$  is the regressor constructed by known nonlinear functions of the generalized coordinates and its first and second time derivatives and  $\boldsymbol{\theta} \in \mathbb{R}^p$  is the vector of  $p$  unknown parameters.

Then for an arbitrary smooth (at least once differentiable) signal  $\dot{\mathbf{q}}_r \in \mathbb{R}^n$ , there should exist a modified regressor  $Y_r(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r) : \mathbb{R}^P \mapsto \mathbb{R}^n$  such that

$$H(\mathbf{q})\ddot{\mathbf{q}}_r + C(\cdot)\dot{\mathbf{q}}_r + D(\cdot)\dot{\mathbf{q}}_r + \mathbf{g}(\mathbf{q}) = Y_r(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\boldsymbol{\theta}. \quad (15)$$

The difference between the estimate version and the real parameters  $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}$  produces an estimate system error:

$$\begin{aligned} Y_r(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\tilde{\boldsymbol{\theta}} &= [H(\mathbf{q}) - \hat{H}(\mathbf{q})]\ddot{\mathbf{q}}_r \\ &+ [C(\mathbf{q}, \dot{\mathbf{q}}) - \hat{C}(\mathbf{q}, \dot{\mathbf{q}})]\dot{\mathbf{q}}_r \\ &+ [D(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{r}_R) - \hat{D}(\mathbf{q})]\dot{\mathbf{q}}_r, \end{aligned} \quad (16)$$

which after the above equivalence is properly bounded:

$$\|Y_r(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\tilde{\boldsymbol{\theta}}\| \leq b_9 \|\dot{\mathbf{q}}_r\| + b_{10} \|\dot{\mathbf{q}}_r\|^2 + b_{11} \|\ddot{\mathbf{q}}_r\|. \quad (17)$$

Then the closed-loop dynamics is found using control law (11) in the open-loop Lagrangian expression (9):

$$\begin{aligned} \hat{H}(\mathbf{q})\dot{\mathbf{s}} + \hat{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{s} + \hat{D}(\mathbf{q})\mathbf{s} + K_s\mathbf{s} + K_i \int \mathbf{s} dt \\ = -\beta \|\mathbf{s}\|^2 \mathbf{s} - Y_r(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\tilde{\boldsymbol{\theta}} - \mathbf{g}(\mathbf{q}) + \boldsymbol{\eta}(\cdot). \end{aligned} \quad (18)$$

Now consider the following Lyapunov candidate function:

$$V(\mathbf{s}) = \frac{1}{2} \mathbf{s}^T \hat{H}(\mathbf{q}) \mathbf{s} + \frac{1}{2} \tilde{\mathbf{a}}^T K_i^{-1} \tilde{\mathbf{a}} > 0, \quad (19)$$

with  $\tilde{\mathbf{a}} \triangleq \mathbf{a}_0 - K_i \int \mathbf{s} dt$  for some constant vector  $\mathbf{a}_0 \in \mathbb{R}^n$ . The time derivative of the Lyapunov candidate function along the trajectories of the closed-loop system (18), after property (13) and proper simplifications, becomes

$$\begin{aligned} \dot{V}(\mathbf{s}) &= -\mathbf{s}^T [\hat{H}(\mathbf{q}) + K_s] \mathbf{s} - \beta \mathbf{s}^T \|\mathbf{s}\|^2 \mathbf{s} - \mathbf{s}^T \mathbf{a}_0 \\ &+ \mathbf{s}^T (\boldsymbol{\eta}(\cdot) - Y_r(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\tilde{\boldsymbol{\theta}} - \mathbf{g}(\mathbf{q})). \end{aligned} \quad (20)$$

Assuming that  $\mathbf{r}_r$  is bounded implies that both  $\dot{\mathbf{q}}_r$  and  $\ddot{\mathbf{q}}_r$  are also bounded. Then, assuming that  $\boldsymbol{\zeta}$  and  $\dot{\boldsymbol{\zeta}}$  are also bounded it yields  $\|\boldsymbol{\eta}(\cdot)\| + \|Y_r(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\tilde{\boldsymbol{\theta}}\| \leq k_1 + k_1 \|\dot{\mathbf{q}}_r\| + k_2 \|\ddot{\mathbf{q}}_r\|^2$ , which can be expressed in terms of the extended error as

$$\|\boldsymbol{\eta}(\cdot)\| + \|Y_r(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\tilde{\boldsymbol{\theta}}\| \leq \mu_0 + \mu_1 \|\mathbf{s}\| + \mu_2 \|\mathbf{s}\|^2. \quad (21)$$

Then the last term in (20) is bounded as follows:

$$\begin{aligned} &\|\boldsymbol{\eta}(\cdot) - Y_r(\cdot)\tilde{\boldsymbol{\theta}} - \mathbf{g}(\mathbf{q})\| \\ &\leq \|\boldsymbol{\eta}(\cdot)\| + \|Y_r(\cdot)\tilde{\boldsymbol{\theta}}\| + \|\mathbf{g}(\mathbf{q})\| \\ &\leq \mu_0 + b_5 + \mu_1 \|\mathbf{s}\| + \mu_2 \|\mathbf{s}\|^2. \end{aligned} \quad (22)$$

Also, let  $\mathbf{a}_0 \triangleq (\mu_0 + b_5)\mathbf{e}_6$ , where  $\mathbf{e}_6 \in \mathbb{R}^6$  is a vector of ones, such that  $\|\mathbf{a}_0\| = (\mu_0 + b_5) > 0$ . Then, after these bounding expressions, (20) is bounded as follows:

$$\dot{V}(\mathbf{s}) \leq -\lambda_{DK} \|\mathbf{s}\|^2 - \beta \|\mathbf{s}\|^4 + \mu_2 \|\mathbf{s}\|^3 + \mu_1 \|\mathbf{s}\|^2, \quad (23)$$

where  $\lambda_{DK}$  is the largest eigenvalue of matrix  $\hat{D}(\mathbf{q}) + K_s$ . The conditions to satisfy  $\dot{V}(\mathbf{s}) < 0$  are summarized as

$$\begin{aligned} \lambda_{DK} &> \mu_1 + \mu_2, \\ \beta &> \mu_2, \end{aligned} \quad (24)$$

which are conditions in the control law gains choice.

Under these conditions  $\dot{V}(\mathbf{s})$  is negative definite and the extended error is asymptotically stable:

$$\lim_{t \rightarrow \infty} \|\mathbf{s}\| \rightarrow 0. \quad (25)$$

Finally, after definition (6) whenever  $\mathbf{s} = 0$  it follows that  $\dot{\mathbf{q}} = -\Lambda \tilde{\mathbf{q}}$  which means that  $\mathbf{q}$  reaches the set point  $\mathbf{q}_d$ . Therefore, the stability for the system is proved. A detailed explanation and analysis of the controller can be found in [33].

The implementation of the control does not require knowledge of the dynamic model parameters; hence it is a robust control with respect to the fluid disturbances and dynamic parametric knowledge. However it is necessary to know the relationship between the control input and the actuators.

**3.2. Thrusters Force Distribution.** The propulsion forces of Mexibot are generated by a set of six fins which move along a sinusoidal path defined as

$$\gamma(t) = \frac{A}{2} \sin\left(\frac{2\pi}{P}t + \delta\right) + \lambda, \quad (26)$$

where  $\gamma$  is the angle of the position of the flip,  $A$  is the amplitude of motion,  $P$  is the period of each cycle,  $\lambda$  is the central angle of the oscillation, and  $\delta$  is the phase offset between different fins of the robot.

Both Georgiades in [35] and Plamondon in [36] show models for the thrust generated by the symmetric oscillation of the fins used in the Aqua robot family. Plamondon presents a relationship between the thrust generated by the fins and the parameters describing the motion in (26). Thus, the magnitude of force generated by each flip with the sinusoidal movement (26) is determined by the following equation:

$$T = 0.1963 \frac{(w_1 + 2w_2) l^2}{3} \rho \frac{A}{P} - 0.1554, \quad (27)$$

where  $l$ ,  $w_1$ , and  $w_2$  correspond to the dimensions of the fins,  $\rho$  represents the density of water,  $A$  is the amplitude, and  $P$  is the period of oscillation. Thus, the magnitude of the force generated by the robot fins can be established in function of the period and the amplitude of the fin oscillation movement at runtime. Figure 8 shows the force produced by the fins, where  $\lambda$  defines the direction and  $T$  the magnitude of the force vector expressed in the body reference frame as

$$\mathbf{F}_p = \begin{bmatrix} F_{px} \\ F_{py} \\ F_{pz} \end{bmatrix}. \quad (28)$$

In addition, due to the kinematic characteristics of the vehicle,  $F_{py} = 0$ . Therefore, the vector of forces and moments generated by the actuators is defined as follows:

$$\mathbf{F}_u = \begin{bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{bmatrix}. \quad (29)$$

Consider the fins numeration as shown in Figure 9; then the following equations state the relationship between the coordinates  $F_{px_j}$  and  $F_{pz_j}$  of  $\mathbf{F}_{p_j}$  and the vector  $\mathbf{F}_u$  as

$$F_x = F_{px_1} + F_{px_2} + F_{px_3} + F_{px_4} + F_{px_5} + F_{px_6} \quad (30a)$$

$$F_y = 0 \quad (30b)$$

$$F_z = F_{pz_1} + F_{pz_2} + F_{pz_3} + F_{pz_4} + F_{pz_5} + F_{pz_6} \quad (30c)$$

$$M_x = l_{y_1} F_{pz_1} + l_{y_2} F_{pz_2} + l_{y_3} F_{pz_3} + l_{y_4} F_{pz_4} + l_{y_5} F_{pz_5} + l_{y_6} F_{pz_6} \quad (30d)$$

$$M_y = l_{x_1} F_{pz_1} + l_{x_2} F_{pz_2} + l_{x_3} F_{pz_3} + l_{x_4} F_{pz_4} + l_{x_5} F_{pz_5} + l_{x_6} F_{pz_6} \quad (30e)$$

$$M_z = l_{y_1} F_{px_1} + l_{y_2} F_{px_2} + l_{y_3} F_{px_3} + l_{y_4} F_{px_4} + l_{y_5} F_{px_5} + l_{y_6} F_{px_6}, \quad (30f)$$

where  $l_{x_j}$  and  $l_{y_j}$  are the distance coordinates of the  $j$ th fin joint with respect to the vehicle's center of mass as shown

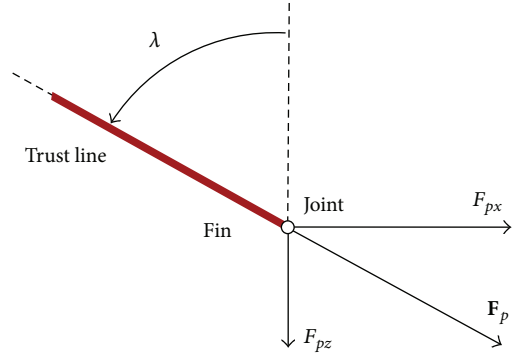


FIGURE 8: Diagram of forces generated by the fins movements where the angle  $\lambda$  establishes the direction of the force.

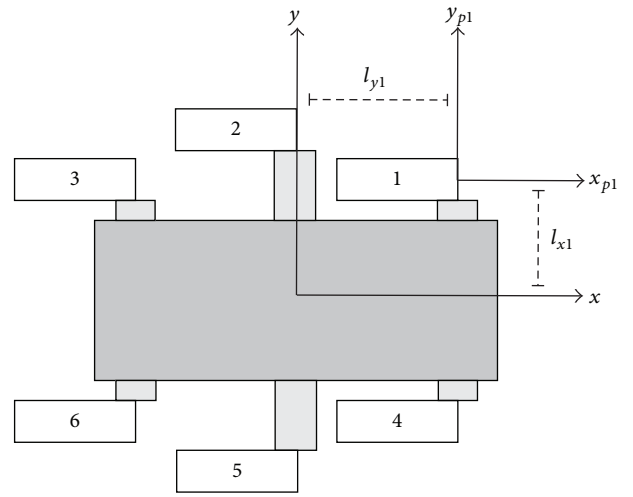


FIGURE 9: Fins distribution in the underwater vehicle.

in Figure 9. Note that the symmetry of the vehicle establishes that  $l_{y_1} = l_{y_3} = -l_{y_4} = -l_{y_6}$ ,  $l_{y_2} = -l_{y_5}$ ,  $l_{x_1} = -l_{x_3} = l_{x_4} = -l_{x_6}$ , and  $l_{x_2} = l_{x_5} = 0$ .

System (30a), (30b), (30c), (30d), (30e), and (30f) has five equations with twelve independent variables. Among all possible solutions the one presented in this work arises after the imposition of the following constraints:

$$\begin{aligned} \text{C1: } & F_{px_1} = F_{px_2} = F_{px_3}, \\ \text{C2: } & F_{px_4} = F_{px_5} = F_{px_6}, \\ \text{C3: } & F_{pz_1} + F_{pz_3} = -F_{pz_4} - F_{pz_6}, \\ \text{C4: } & F_{pz_1} - F_{pz_3} = F_{pz_4} - F_{pz_6}, \\ \text{C5: } & F_{pz_2} = -F_{pz_5}, \\ \text{C6: } & F_z = 0. \end{aligned} \quad (31)$$

Then one system solution is found to be

$$\begin{bmatrix} F_{px_1} \\ F_{px_4} \\ F_{py_1} \\ F_{py_2} \\ F_{py_3} \\ F_{py_4} \\ F_{py_5} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \sigma_1 \\ \frac{1}{2} & 0 & 0 & -\sigma_1 \\ 0 & \sigma_1 & \sigma_2 & 0 \\ 0 & \sigma_1 & 0 & 0 \\ 0 & \sigma_1 & -\sigma_2 & 0 \\ 0 & -\sigma_1 & \sigma_2 & 0 \\ 0 & -\sigma_1 & -\sigma_2 & 0 \end{bmatrix} \begin{bmatrix} F_x \\ M_x \\ M_y \\ M_z \end{bmatrix}, \quad (32)$$

where

$$\begin{aligned} \sigma_1 &= \frac{1}{2(2l_{y_1} + l_{y_2})}, \\ \sigma_2 &= \frac{1}{4l_{x_1}}. \end{aligned} \quad (33)$$

Now, the oscillation amplitude  $A_j$  of the  $j$ th fin is computed after (27) using an oscillation period of 0.4 [s], and the corresponding thrust is defined as

$$T_i = \sqrt{F_{px_j}^2 + F_{pz_j}^2}. \quad (34)$$

Finally, the central angle of oscillation is computed as

$$\lambda_j = \tan^{-1} \left( \frac{F_{px_j}}{F_{pz_j}} \right). \quad (35)$$

**3.3. Desired Signals Computation.** In this navigation system the controller performs a set-point task. The desired values are computed based on the visual information. Due to the under-actuated nature of the vehicle and sensor limitations, only the attitude and depth of the vehicle can be controlled. The desired depth value  $z_d$  is a constant and the roll desired angle is always  $\phi_d = 0$ . As the constraint C6:  $F_z = 0$  has been considered, the depth is controlled indirectly by modifying the desired pitch angle  $\theta_d$ . This desired orientation angle is calculated in terms of the depth error as

$$\theta_d = k_z (z - z_d), \quad (36)$$

where  $k_z$  is a positive constant.

The visual system defines the desired yaw angle  $\psi_d$ . Images from the left camera are processed in a ROS node with the algorithms described in Section 2.1 in order to determine the presence of the sphere in the field of view. When visual information is not available, this angle remains constant with the initial value or with the last computed value. However, if a single circle with a radius bigger than a certain threshold is found, the new desired yaw angle is computed based on the visual error. This error is defined as the distance in pixels between the center of the image and the position in the  $x$ -axis of the detected circle. So, the new desired yaw angle is computed as

$$\psi_d = \phi + \tilde{v}_x \frac{300r}{(\text{columns} \times \text{rows})}, \quad (37)$$

where  $\psi$  is the actual yaw angle,  $\tilde{v}_x$  is the visual error in horizontal axis, rows and columns are the image dimensions, and  $r$  is the radius of the circle. This desired yaw angle is proportional to the visual error, but it also depends on the radius of the circle found. When the object is close to the camera, the sphere radius is larger, and therefore the change of  $\psi_d$  also increases. Note that the resolution of the image given by the vehicle's camera is  $320 \times 240$  pixels; with this, the gain used to define the reference yaw angle in (37) was established as 300. This value was obtained experimentally, with a trial error procedure, and produces a correction of approximately  $1^\circ$ , with a visual error  $\tilde{v}_x = 10$  and radius of the observed sphere  $r = 25$ . This update of the desired yaw angle modifies the vehicle attitude and reduces the position error of the sphere in the image. We note that the update of the desired yaw angle was necessary only when the visual error was bigger than 7 pixels; by this reason when  $v_x$  is smaller than this threshold the reference signal keeps the previous value.

Finally, when a circle inside of other circle is found, that means the underwater vehicle is close to the mark and a direction change is performed. The desired yaw angle is set to the actual yaw value plus an increment related to the location of the next sphere. This integration of the visual system and the controller results in the autonomous navigation system for underwater vehicle which is able to track the marks placed in a semistructured environment.

## 4. Experimental Results

In order to evaluate the performance of the visual navigation system, a couple of experimental results are presented in this section. Two red spheres were placed in a pool with turbid water. An example of the type of view in this environment is shown in Figure 10. This environment is semistructured because the floor is not natural and also because of the lack of currents; however, the system is subjected to disturbances produced by the movement of swimmers which closely follow the robot. As it was mentioned before, the exact position of the spheres is unknown; only the approximate angle which relates the position between the marks is available. Figure 11 shows a diagram with the artificial marks distribution. The underwater vehicle starts swimming towards one of the spheres. Although the circle detection algorithm includes functions for selecting the circle of interest when more than one are detected, for this first experiment, no more than one visual mark is in front of the visual field of the camera at the same time.

The implementation of the attitude and depth control was performed in a computer with QNX real-time operating system and the sample time of the controller is 1 ms. This controller accesses the inertial sensors in order to regulate the depth and orientation of the vehicle. The reference signal of the yaw angle was set with the initial orientation of the robot and updated by the visual system when a sphere is detected. This visual system was implemented in a computer with Ubuntu and ROS, having an approximate sample time of 33 ms when a visual mark is present. The parameters



TABLE 1: Parameters of our AUV used in the experimental test.

Notation	Description	Value	Units
$\bar{m}$	Mexibot mass	1.79	[kg]
$\hat{I}_{xx}$	Inertia moment with respect to $x$ -axis	0.001	[kg m <sup>2</sup> ]
$\hat{I}_{yy}$	Inertia moment with respect to $y$ -axis	0.001	[kg m <sup>2</sup> ]
$\hat{I}_{zz}$	Inertia moment with respect to $z$ -axis	0.001	[kg m <sup>2</sup> ]
$l_{x_1}$	Distance between the AUV center of mass and the position of the fin 1 in axis $x$	0.263	[m]
$l_{y_1}$	Distance between the AUV center of mass and the position of the fin 1 in axis $y$	0.149	[m]
$l_{y_2}$	Distance between the AUV center of mass and the position of the fin 2 in axis $y$	0.199	[m]
$l$	Fin length	0.190	[m]
$w_1$	Fin width 1	0.049	[m]
$w_2$	Fin width 2	0.068	[m]
$\rho$	Water density	1000	[kg/m <sup>3</sup> ]

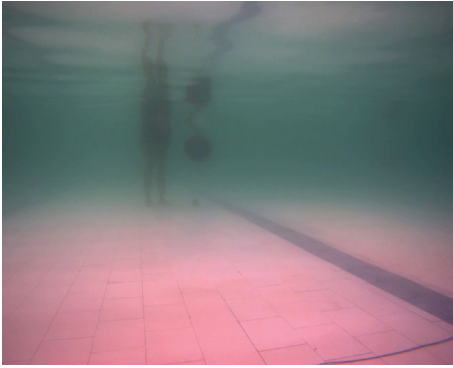


FIGURE 10: Diagram to illustrate the approximate location of visual marks.

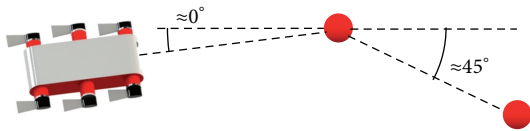


FIGURE 11: Diagram to illustrate the approximate location of visual marks.

used in the implementation are presented in Table 1 and were obtained from the manufacturer. Water density value has been used with a nominal value assuming that the controller is able to handle the inaccuracy with respect to real values.

The control gains in (4) and (36) were established after a trial and error process. The nonlinear and strongly coupled

nature of the vehicle dynamics causes the fact that small variations in the control gains affect considerably the performance of the controller. For the experimental validation, we first tuned the gains of the attitude controller, following this sequence:  $K_s$ ,  $\Lambda$ ,  $\beta$ , and  $K_i$ . Then, the parameter  $k_z$  of the depth controller was selected. With this, the control gains were set as follows:

$$\begin{aligned}
 \beta &= 0.7, \\
 k_z &= 17, \\
 K_s &= \text{diag}\{0.30, 0.20, 1.50\}, \\
 K_i &= \text{diag}\{0.10, 0.07, 0.20\}, \\
 \Lambda &= \text{diag}\{0.50, 0.50, 0.50\}.
 \end{aligned} \tag{38}$$

In the first experiment, the navigation task considers the following scenario. A single red sphere is placed in front of the vehicle approximately at 8 meters of distance. The time evolution of the depth coordinate  $z(t)$  and the attitude signals  $\phi(t)$ ,  $\theta(t)$ ,  $\psi(t)$  are shown in Figure 12, where the corresponding depth and attitude reference signals are also plotted. The first 20 seconds corresponds to the start-up period of the navigation system. After that, the inertial controller ensures that the vehicle moves in the same direction until the navigation system receives visual feedback. This feedback occurs past thirty seconds and the desired value for the angle yaw  $\psi(t)$  starts to change in order to follow the red sphere. Notice that the reference signal for the pitch angle  $\theta_d(t)$  presents continuous changes after the initialization period. This is because the depth control is performed indirectly by modifying the value of  $\theta_d$  with (36). In addition the initial value for  $\psi_d(t)$  is not relevant because

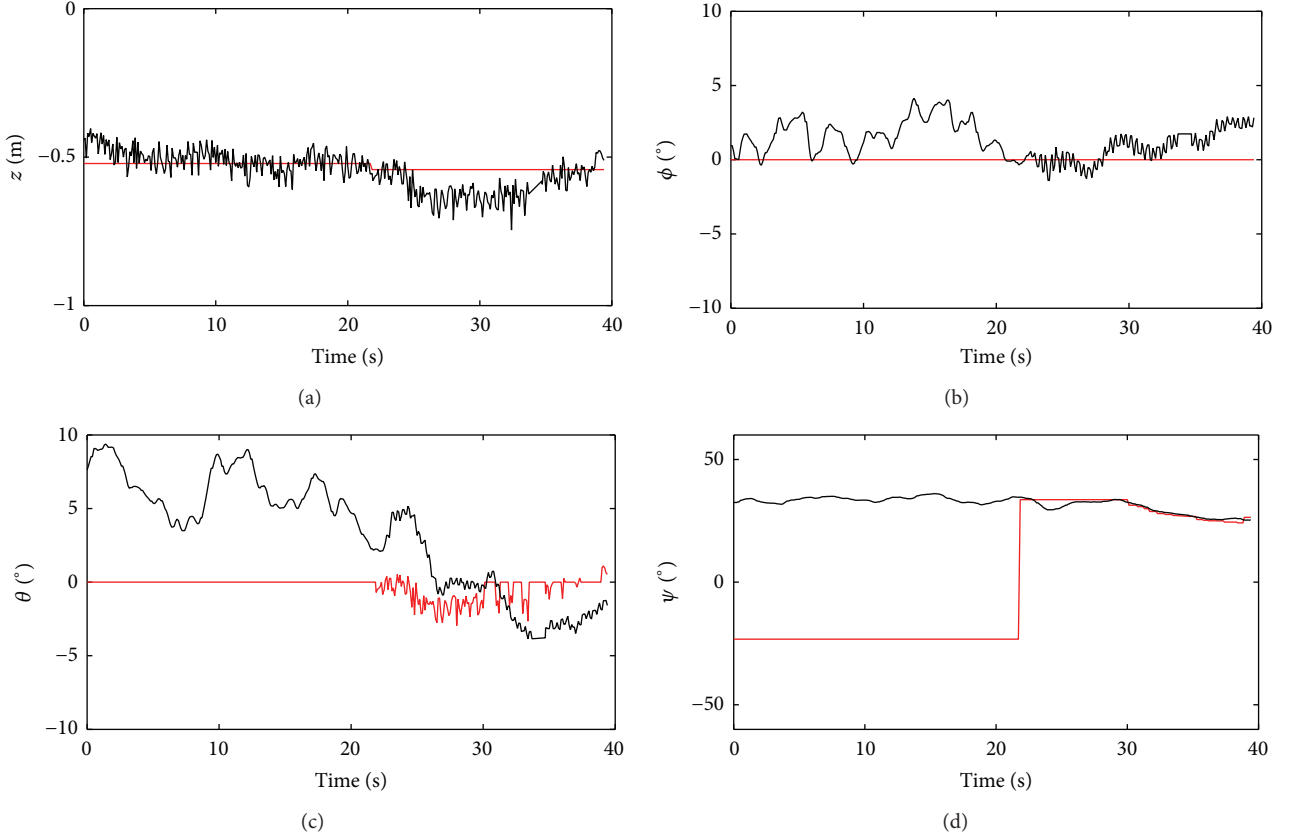


FIGURE 12: Navigation experiment when tracking one sphere. Desired values in red and actual values in black. (a) Depth  $z(t)$ ; (b) roll  $\phi(t)$ ; (c) pitch  $\theta(t)$ ; (d) yaw  $\psi(t)$ .

this value is updated after the inertial navigation system starts. The corresponding depth and attitude error signals are depicted in Figure 13, where all of these errors have considerably small magnitude, bounded by a value around 0.2 m for the depth error and  $10^{\circ}$  for attitude error.

The time evolution of the visual error in the horizontal axis is depicted in Figure 14. Again, the first thirty seconds does not show relevant information because no visual feedback is obtained. Later, the visual error is reduced to a value in an acceptable interval represented by the red lines. This interval represents the values where the desired yaw angle does not change, even when the visual system is not detecting the sphere. As mentioned before, the experiments show that when  $v_x \leq 7$  pixels, the AUV can achieve the assigned navigation task. Finally, a disturbance, generated by nearby swimmers when they displace water, moves the vehicle and the error increases, but the visual controller acts to reduce this error.

The previous results show that the proposed controller (4) under the thruster force distribution (32) provides a good behavior in the set-point control of the underwater vehicle, with small depth and attitude error values. This performance enables the visual navigation system to track the artificial marks placed in the environment.

The navigation task assigned to the underwater vehicle in the second experiment includes the two spheres with

the distribution showed in Figure 11. For this experiment the exact position of the spheres is unknown; only the approximate relative orientation and distance between them are known. The first sphere was positioned in front of the AUV at an approximated distance of 8 m. When the robot detects that the first ball was close enough, it should change the yaw angle to  $45^{\circ}$  in order to find the second sphere. Figure 16 shows the time evolution of the depth coordinate  $z(t)$ , the attitude signals  $\phi(t)$ ,  $\theta(t)$ ,  $\psi(t)$ , and the corresponding reference signals during the experiment. Similarly to the previous experiment, the actual depth, roll angle, and pitch angle are close to the desired value, even when small ambient disturbances are present. The yaw angle plot shows the different stages of the system. The desired value at the starting period is an arbitrary value that does not have any relation with the vehicle state. After the initialization period, a new desired value for the yaw angle is set and this angle remains constant as long as the visual system does not provide information. When the visual system detects a sphere, the navigation system generates a smooth desired signal allowing the underwater vehicle to track the artificial mark. When the circle inside of the other circle was detected, the change in direction of  $45^{\circ}$  was applied. This reference value was fixed until the second sphere was detected and a new desired signal was generated with small changes. Finally, the second circle inside of the sphere was detected and a new

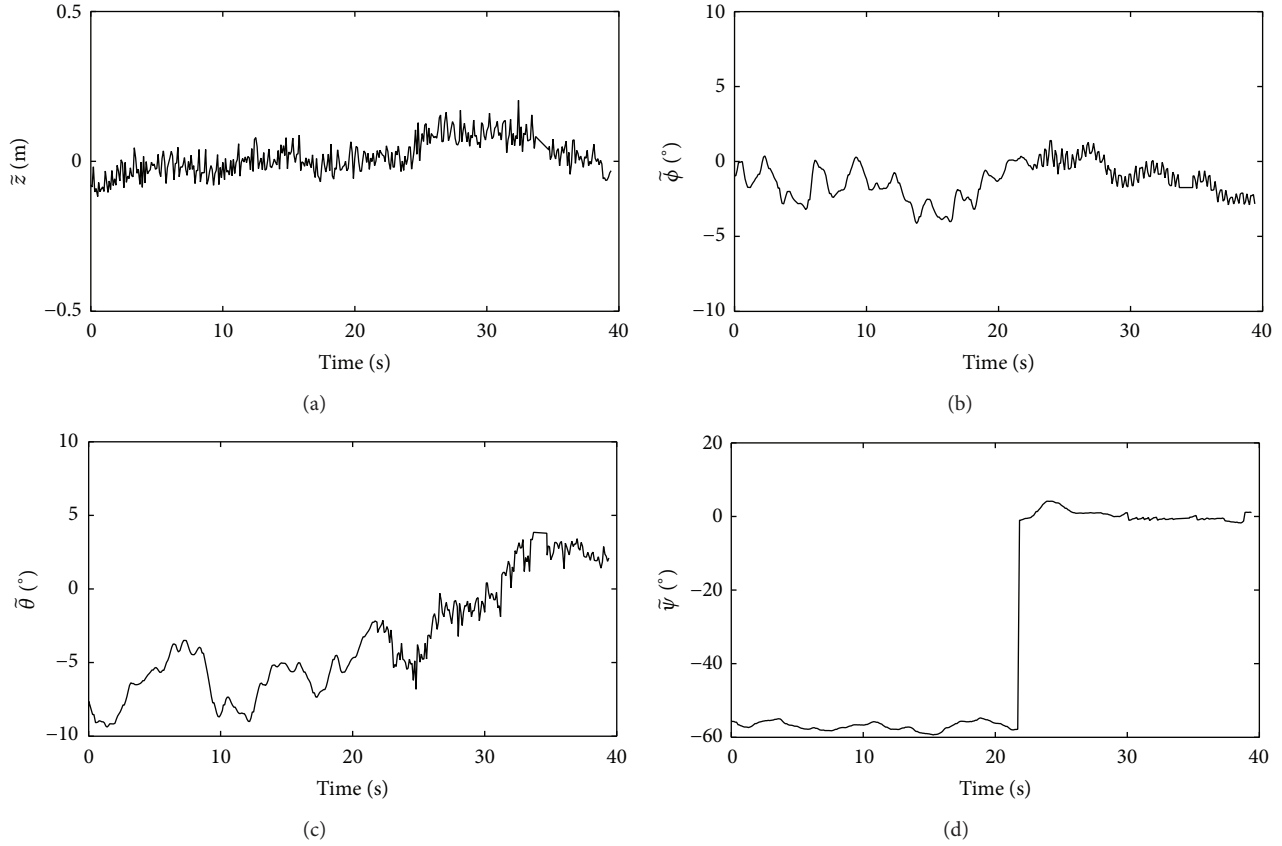


FIGURE 13: Navigation tracking errors of one sphere when using controller (4). (a) Depth  $z(t)$ ; (b) roll  $\phi(t)$ ; (c) pitch  $\theta(t)$ ; (d) yaw  $\psi(t)$ .

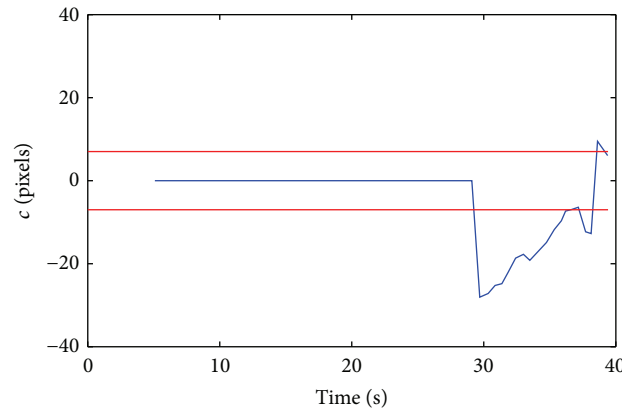


FIGURE 14: Navigation tracking of one sphere. Visual error obtained from the AUV navigation experiment.

change of  $45^\circ$  was performed and the desired value remained constant until the end of the experiment.

Figure 17 shows the depth and attitude error signals. Similar to the first experiment, the magnitude of this error is bounded by a value around 0.2 m for the depth error and  $10^\circ$  for the attitude error, except for the yaw angle, which presents higher values produced by the direction changes. Note that, in this experiment, significant amount of the error was produced by environmental disturbances.

Finally, the graph of the time evolution of the visual error is depicted in Figure 15. It can be observed that, at the beginning, while the robot was moving forward, the error remained constant because the system was unable to determine the presence of the artificial mark in the environment. At a given time  $t_i$ , the visual system detected the first sphere, with an estimated radius  $r_{t_i}$  of about 30 pixels. Then, as the robot gets closer to the target, the visual error begins to decrease due to the improvement in visibility and

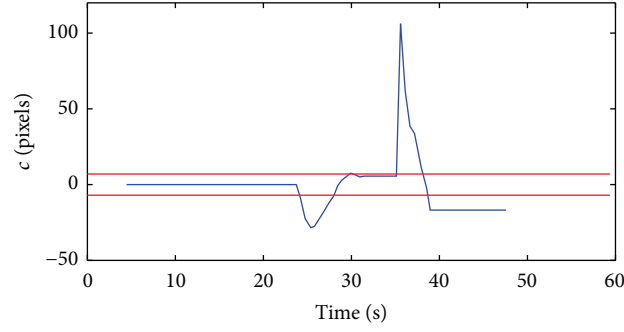


FIGURE 15: Navigation tracking of two spheres. Visual error obtained from the AUV navigation experiment.

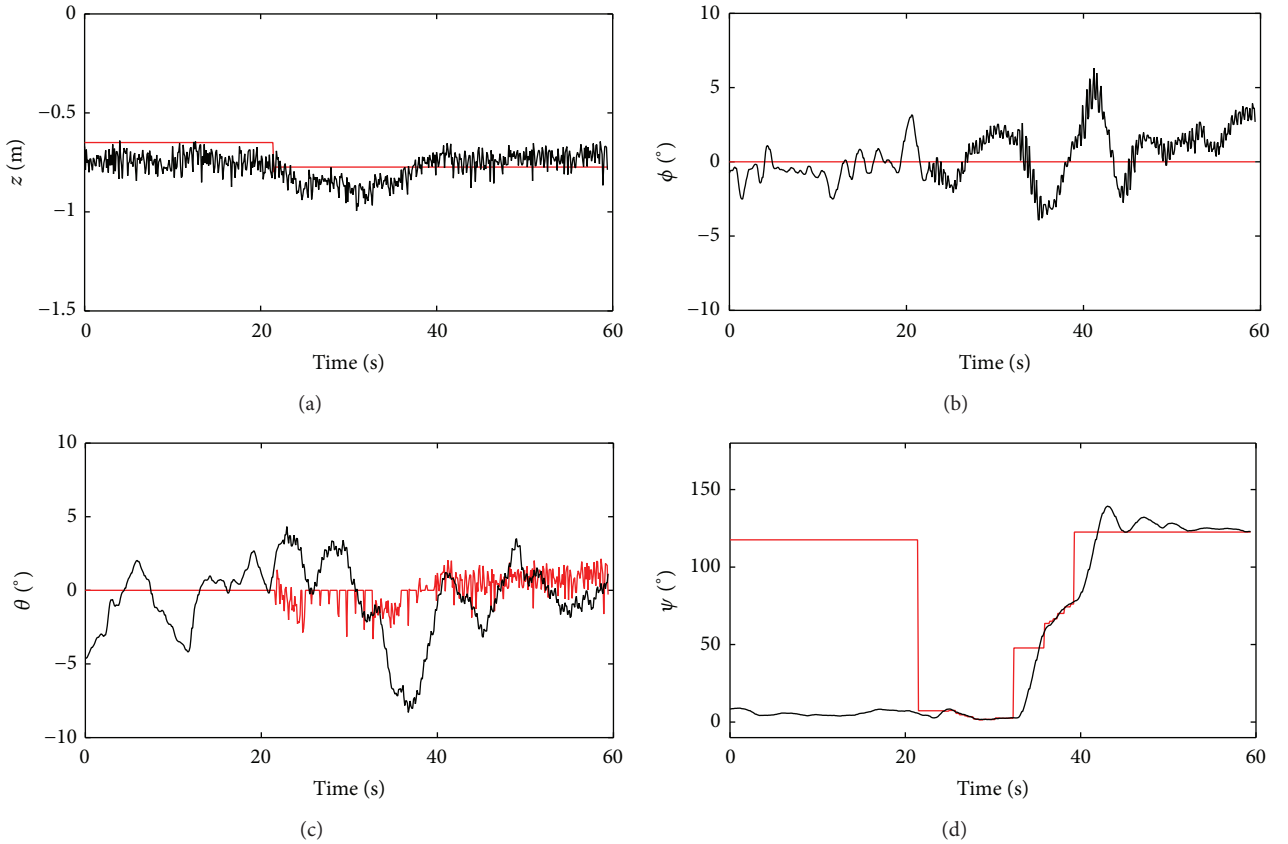


FIGURE 16: Navigation tracking of two spheres. Desired values in red and actual values in black. (a) Depth  $z(t)$ ; (b) roll  $\phi(t)$ ; (c) pitch  $\theta(t)$ ; (d) yaw  $\psi(t)$ .

the radius of the sphere increases. When the radius is bigger than a given threshold, a change-of-direction action is fired in order to avoid collision and to search for the second sphere. Then, all variables are reset. Once again the error remains constant at the beginning due to the lack of visual feedback. In this experiment, when the second mark was identified, the visual error was bigger than 100 pixels, but rapidly this error decreased to the desired interval. At the end of the experiment, another change of direction was generated and the error remained constant, because no other sphere in the environment was detected.

## 5. Conclusion

In this paper, a visual-based controller to guide the navigation of an AUV in a semistructured environment using artificial marks was presented. The main objective of this work is to provide to an aquatic robot the capability of moving in an environment when visibility conditions are far from ideal and artificial landmarks are placed with an approximately known distribution. A robust control scheme applied under a given thruster force distribution combined with a visual servoing control was implemented. Experimental evaluations



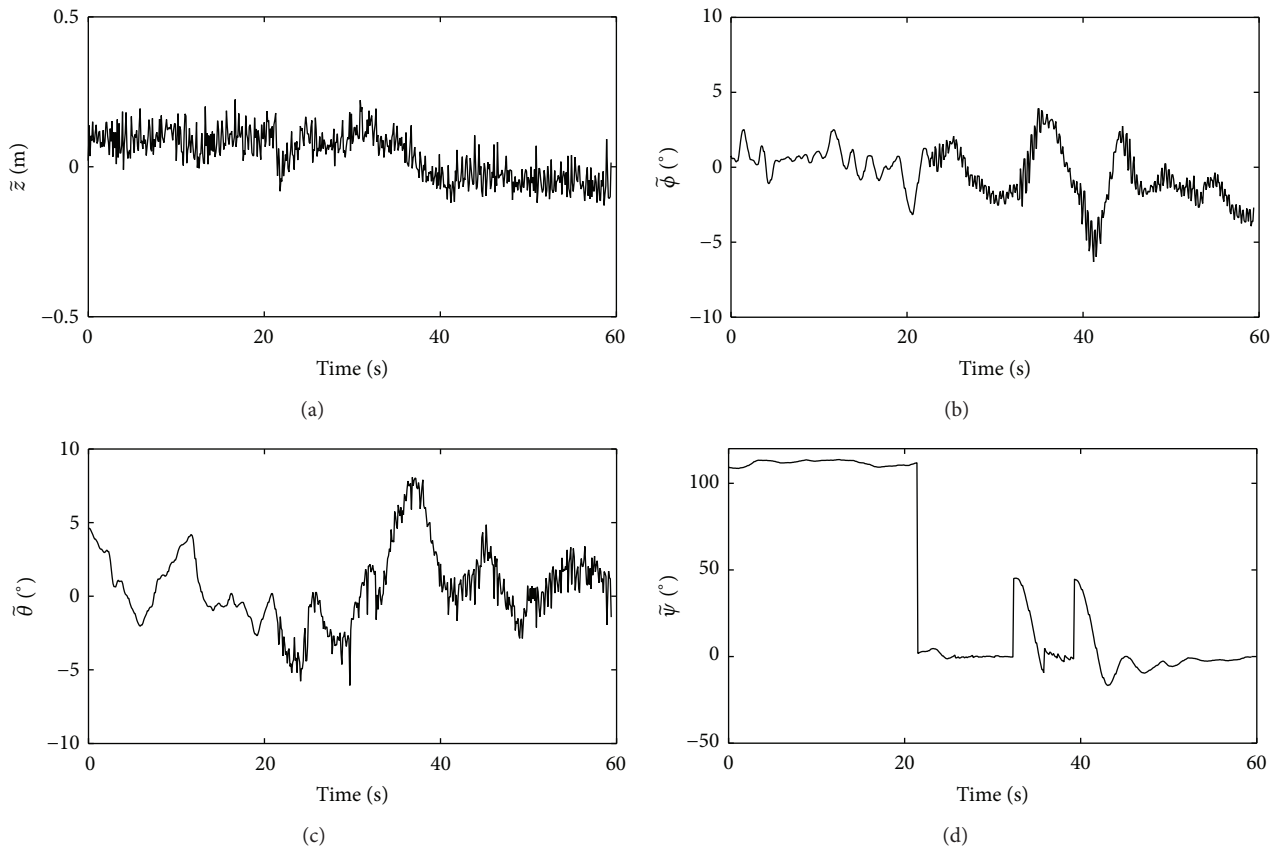


FIGURE 17: Navigation error when tracking two spheres when using controller (4). (a) Depth  $z(t)$ ; (b) roll  $\phi(t)$ ; (c) pitch  $\theta(t)$ ; (d) yaw  $\psi(t)$ .

for the navigation system were carried out in an aquatic environment with poor visibility. The results show that our approach was able to detect the visual marks and perform the navigation satisfactorily. Future work includes the use of natural landmarks and to lose some restrictions, for example, that more than one visual mark can be present in the field of view of the robot.

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

The authors thank the financial support of CONACYT, México.

## References

- [1] J. J. Leonard, A. A. Bennett, C. M. Smith, and H. J. S. Feder, "Autonomous underwater vehicle navigation," in *Proceedings of the IEEE ICRA Workshop on Navigation of Outdoor Autonomous Vehicles*, 1998.
- [2] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "A survey of underwater vehicle navigation: recent advances and new challenges," in *Proceedings of the IFAC Conference of Manoeuvring and Control of Marine Craft*, vol. 88, 2006.
- [3] L. Stutters, H. Liu, C. Tiltman, and D. J. Brown, "Navigation technologies for autonomous underwater vehicles," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 38, no. 4, pp. 581–589, 2008.
- [4] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: a review," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014.
- [5] A. Hanai, S. K. Choi, and J. Yuh, "A new approach to a laser ranger for underwater robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '03)*, pp. 824–829, October 2003.
- [6] F. R. Dalgleish, F. M. Caimi, W. B. Britton, and C. F. Andren, "An AUV-deployable pulsed laser line scan (PLLS) imaging sensor," in *Proceedings of the MTS/IEEE Conference (OCEANS '07)*, pp. 1–5, Vancouver, Canada, September 2007.
- [7] A. Annunziatellis, S. Graziani, S. Lombardi, C. Petrioli, and R. Petrocchia, "CO<sub>2</sub>Net: a marine monitoring system for CO<sub>2</sub> leakage detection," in *Proceedings of the OCEANS, 2012*, pp. 1–7, IEEE, Yeosu, Republic of Korea, 2012.
- [8] G. Antonelli, *Underwater Robots-Motion and Force Control of Vehicle-Manipulator System*, Springer, New York, NY, USA, 2nd edition, 2006.
- [9] T. Nicosevici, R. Garcia, M. Carreras, and M. Villanueva, "A review of sensor fusion techniques for underwater vehicle navigation," in *Proceedings of the MTTs/IEEE TECHNO-OCEAN '04 (OCEANS '04)*, vol. 3, pp. 1600–1605, IEEE, Kobe, Japan, 2004.

- [10] F. Bonin-Font, G. Oliver, S. Wirth, M. Massot, P. L. Negre, and J.-P. Beltran, "Visual sensing for autonomous underwater exploration and intervention tasks," *Ocean Engineering*, vol. 93, pp. 25–44, 2015.
- [11] K. Teo, B. Goh, and O. K. Chai, "Fuzzy docking guidance using augmented navigation system on an AUV," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 2, pp. 349–361, 2015.
- [12] R. B. Wynn, V. A. I. Huvenne, T. P. Le Bas et al., "Autonomous Underwater Vehicles (AUVs): their past, present and future contributions to the advancement of marine geoscience," *Marine Geology*, vol. 352, pp. 451–468, 2014.
- [13] F. Bonin-Font, M. Massot-Campos, P. L. Negre-Carrasco, G. Oliver-Codina, and J. P. Beltran, "Inertial sensor self-calibration in a visually-aided navigation approach for a micro-AUV," *Sensors*, vol. 15, no. 1, pp. 1825–1860, 2015.
- [14] J. Santos-Victor and J. Senteiro, "The role of vision for underwater vehicles," in *Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology (AUV '94)*, pp. 28–35, IEEE, Cambridge, Mass, USA, July 1994.
- [15] A. Burguera, F. Bonin-Font, and G. Oliver, "Trajectory-based visual localization in underwater surveying missions," *Sensors*, vol. 15, no. 1, pp. 1708–1735, 2015.
- [16] D. Kim, D. Lee, H. Myung, and H.-T. Choi, "Artificial landmark-based underwater localization for AUVs using weighted template matching," *Intelligent Service Robotics*, vol. 7, no. 3, pp. 175–184, 2014.
- [17] J. Sattar and G. Dudek, "Robust servo-control for underwater robots using banks of visual filters," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 3583–3588, Kobe, Japan, May 2009.
- [18] C. Barngrover, S. Belongie, and R. Kastner, "Jboost optimization of color detectors for autonomous underwater vehicle navigation," in *Computer Analysis of Images and Patterns*, pp. 155–162, Springer, 2011.
- [19] J. Gao, A. Proctor, and C. Bradley, "Adaptive neural network visual servo control for dynamic positioning of underwater vehicles," *Neurocomputing*, vol. 167, pp. 604–613, 2015.
- [20] S. Heshmati-Alamdari, A. Eqtami, G. C. Karras, D. V. Dimarogonas, and K. J. Kyriakopoulos, "A self-triggered visual servoing model predictive control scheme for under-actuated underwater robotic vehicles," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '14)*, pp. 3826–3831, Hong Kong, June 2014.
- [21] P. L. N. Carrasco, F. Bonin-Font, and G. O. Codina, "Stereo graph-slam for autonomous underwater vehicles," in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems*, pp. 351–360, 2014.
- [22] B. Li, Y. Xu, C. Liu, S. Fan, and W. Xu, "Terminal navigation and control for docking an underactuated autonomous underwater vehicle," in *Proceedings of the IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER '15)*, pp. 25–30, Shenyang, China, June 2015.
- [23] M. Myint, K. Yonemori, A. Yanou, S. Ishiyama, and M. Minami, "Robustness of visual-servo against air bubble disturbance of underwater vehicle system using three-dimensional marker and dual-eye cameras," in *Proceedings of the MTS/IEEE Washington (OCEANS '15)*, pp. 1–8, IEEE, Washington, DC, USA, 2015.
- [24] B. Sütő, R. Dóczy, J. Kalló et al., "HSV color space based buoy detection module for autonomous underwater vehicles," in *Proceedings of the 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI '15)*, pp. 329–332, IEEE, Budapest, Hungary, November 2015.
- [25] M. Bryson, M. Johnson-Roberson, O. Pizarro, and S. B. Williams, "True color correction of autonomous underwater vehicle imagery," *Journal of Field Robotics*, 2015.
- [26] A. Yamashita, M. Fujii, and T. Kaneko, "Color registration of underwater images for underwater sensing with consideration of light attenuation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 4570–4575, Roma, Italy, April 2007.
- [27] D. L. Ruderman, T. W. Cronin, and C.-C. Chiao, "Statistics of cone responses to natural images: implications for visual coding," *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, vol. 15, no. 8, pp. 2036–2045, 1998.
- [28] T. D'Orazio, C. Guaragnella, M. Leo, and A. Distanto, "A new algorithm for ball recognition using circle hough transform and neural classifier," *Pattern Recognition*, vol. 37, no. 3, pp. 393–408, 2004.
- [29] C. Akinlar and C. Topal, "EDCircles: a real-time circle detector with a false detection control," *Pattern Recognition*, vol. 46, no. 3, pp. 725–740, 2013.
- [30] G. Dudek, P. Giguere, C. Prahacs et al., "AQUA: an amphibious autonomous robot," *Computer*, vol. 40, no. 1, pp. 46–53, 2007.
- [31] U. Saranli, M. Buehler, and D. E. Koditschek, "RHex: a simple and highly mobile hexapod robot," *International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, 2001.
- [32] T. I. Fossen, *Guidance and Control of Ocean Vehicles*, John Wiley & Sons, 1994.
- [33] R. Pérez-Alcocer, E. Olguín-Díaz, and L. A. Torres-Méndez, "Model-free robust control for fluid disturbed underwater vehicles," in *Intelligent Robotics and Applications*, C.-Y. Su, S. Rakheja, and H. Liu, Eds., vol. 7507 of *Lecture Notes in Computer Science*, pp. 519–529, Springer, Berlin, Germany, 2012.
- [34] E. Olguín-Díaz and V. Parra-Vega, "Tracking of constrained submarine robot arms," in *Informatics in Control, Automation and Robotics*, vol. 24, pp. 207–222, Springer, Berlin, Germany, 2009.
- [35] C. Georgiades, *Simulation and control of an underwater hexapod robot [M.S. thesis]*, Department of Mechanical Engineering, McGill University, Montreal, Canada, 2005.
- [36] N. Plamondon, *Modeling and control of a biomimetic underwater vehicle [Ph.D. thesis]*, Department of Mechanical Engineering, McGill University, Montreal, Canada, 2011.

## Research Article

# Indoor Positioning System Using Depth Maps and Wireless Networks

**Jaime Duque Domingo,<sup>1</sup> Carlos Cerrada,<sup>1</sup> Enrique Valero,<sup>2</sup> and J. A. Cerrada<sup>1</sup>**

<sup>1</sup>*Departamento de Ingeniería de Software y Sistemas Informáticos, ETSI Informática, UNED, C/Juan del Rosal 16, 28040 Madrid, Spain*

<sup>2</sup>*School of Energy, Geoscience, Infrastructure and Society, Heriot-Watt University, Edinburgh EH14 4AS, UK*

Correspondence should be addressed to Jaime Duque Domingo; [jaimeduque@amenofis.com](mailto:jaimeduque@amenofis.com)

Received 11 March 2016; Revised 18 May 2016; Accepted 23 May 2016

Academic Editor: Juan A. Corrales

Copyright © 2016 Jaime Duque Domingo 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 work presents a new *Indoor Positioning System* (IPS) based on the combination of *WiFi Positioning System* (WPS) and *depth maps*, for estimating the location of people. The combination of both technologies improves the efficiency of existing methods, based uniquely on wireless positioning techniques. While other positioning systems force users to wear special devices, the system proposed in this paper just requires the use of *smartphones*, besides the installation of RGB-D sensors in the sensing area. Furthermore, the system is not intrusive, being not necessary to know people's identity. The paper exposes the method developed for putting together and exploiting both types of sensory information with positioning purposes: the measurements of the level of the signal received from different access points (APs) of the wireless network and the *depth maps* provided by the RGB-D cameras. The obtained results show a significant improvement in terms of positioning with respect to common WiFi-based systems.

## 1. Introduction

*Indoor Positioning Systems* (IPS) are techniques used to obtain the position of people or objects inside a building [1]. Among these, *WiFi Positioning Systems* (WPS) [2] are those based on portable devices, such as *cell phones*, to locate people or objects by means of the measurements of the level of the signal received from different access points (APs), that is, WiFi routers.

In the field of people and objects detection, other technologies, such as those based on artificial vision, have been increasingly used. In fact, object recognition can be considered as a part of the core research area of artificial vision, and an important number of authors have reported methods and applications for people detection and positioning. More recent and therefore less abundant are those works involving the use of modern technologies such as RGB-D sensors, which provide 3D information in form of *depth maps* of scenes. For example, Saputra et al. [3] present an indoor human tracking application using 2 depth cameras. Although some effort has been made in applications using the above-mentioned technologies, there is a lack of research on the

combined use of both types of technologies for positioning purposes.

This paper presents a new IPS approach based on the combination of these two different technologies: WPS and *depth maps*, in an active manner. By active combination, these authors mean that the developed method puts together and exploits coordinately both types of sensory information: strength of measured wireless signals and *depth maps*.

This approach is particularly advantageous when several users are simultaneously in a room. In this case, the system is able to detect each user with the help of the coordinates of the people located in a *depth map*. WPS approximates the position of the users, but when they are really close, the proposed method is able to deliver a more precise location. This is carried out with the help of user trajectories, which are considered in two ways: WPS trajectory and trajectory of the people in the *depth map*. As demonstrated in the following sections, this combination improves the efficiency of the existing approaches used in WPS.

The paper is structured as follows: Section 2 explores existing solutions concerning the positioning, based on WPS

and RGB-D sensors, and using both technologies in a joint manner. Section 3 is devoted to describing in detail the basis of the proposed system and how it works. Section 4 presents the performed experiments and analyzes the obtained results. Finally, Section 5 remarks the main advantages of the presented system and shows future developments based on this method.

## 2. Overview of the Related Work

Recently, Subbu et al. [4] established three types of IPS: *fingerprinting*, which uses the signals obtained from portable device such as WiFi, sound, light, or magnetic fields; *crowdsensing*, an extension of *fingerprinting* that continuously updates the positioning database; and finally *Dead Reckoning Systems*, using the accelerometer sensor of portable devices to obtain the inertial movement and the magnetometer to obtain the direction of the magnetic field.

WPS is founded on the *fingerprinting* technique [5], in which a map of the environment is created recording various values of *Received Signal Strength Indication* (RSSI) in each point. RSSI is a reference scale used to measure the power level of signals received from a device on a wireless network (usually WiFi or mobile telephony). This map is used to obtain the position of a user in real time, comparing the values received from the user's portable device to those stored in the map.

Quan et al. [6] show how WPS based on *fingerprint maps* works better than those techniques based on triangulation, like RADAR [7]. This technique [7] records and processes signal strength information at multiple base stations and combines empirical measurements with signal propagation modeling to determine users location by means of the triangulation.

The positioning with *fingerprint map* is carried out in two ways: considering the nearest neighbor, where the Euclidean distances between the live RSSI reading and each reference point fingerprint is calculated for determining the position, and the probabilistic location with Markov, where statistical data of the fingerprint are used to guess the most likely position. The results shown indicate that the nearest neighbor approach works better than the Markov-based one. The triangulation method provides worse results because equations do not transform properly RSSI values into distance, due to the presence of walls and obstacles. Other works have tried to obtain that distance through the use of fuzzy logic [8] or particle filters [9].

Regarding approaches based on fingerprinting, Martin et al. [10] study the accuracy of different techniques: *Closest Point*, *Nearest Neighbor in Signal*, *Average Smallest Polygon*, and *Nearest Neighbor in Signal and Access Point averages*. Depending on the room or cell size where the user is situated, the positioning results are different. The successes are between 78% and 87% determining the room where the user is. If the user is in a room and  $2 \times 2$  meters cells have been created, the successes are between 39% and 48% determining the cell where the user is. When  $1 \times 1$  meter cells are used, the successes are between 18% and 32%.

Considering the distance between APs and receivers, Kornuta et al. [11] analyze the attenuation of the signal produced when the APs are far from the receiver or there are walls or obstacles along the way. Some filters are studied in [12] for attenuating the noise of RSSI. The work [13] studies the combination of WiFi and *Inertial Navigation Systems* (INS) in order to obtain the trajectory of the user. Three sensors are used: gyroscope, accelerometer, and an atmospheric pressure sensor. Husen and Lee [14] propose how to obtain the user orientation with a *fingerprint map*.

In the field of people and objects detection, other technologies, such as those based on artificial vision (e.g., RGB-D sensors), have been increasingly used. Ye et al. [15] propose to use three Kinect sensors for detecting and identifying several people that are occluded by others in a scene. In [16], authors propose a *smart-cane* for the visually impaired that, with the help of a Kinect sensor, allows for locating objects. The method *Kinect Positioning System* (KPS) is analyzed in [17] aiming to obtain the user position.

These positioning techniques have also been used in Robotics. A noteworthy example can be found in [18], where several *Simultaneous Localization and Mapping* (SLAM) algorithms are proposed for building maps using robots with continuous positioning. Mirowski et al. [19] analyze how to generate a *fingerprint map* with an RGB-D sensor mounted on a robot. By means of SLAM, the environment is built recording the measurements RSSI in each point. Also, in this field of research, the use of distinct technologies allows for improving the positioning systems. In [20], a robot is located using three different systems: a laser rangefinder, a depth camera, and the RSSI values. Each system is used independently according to the zone where the robot is located.

RFID techniques have been proposed for location and tracking of users inside buildings as presented in [21], where authors propose to combine identification and positioning based on RFID with the Kinect sensor for obtaining the precise position of a person inside an environment. In this case, one fix RFID reader is located in the room. Each user carries their own RFID tag while the Kinect sensor obtains the skeletons of two people. Each skeleton is composed of the coordinates of the different joints of a person: neck, shoulders, elbows, knees, and so forth. Other methods use RFID tags on the floor where the users can know their positions thanks to a RFID reader they carry with them [22].

However, RFID techniques present several disadvantages, such as interferences with materials and devices, and do not provide too precise location results. These inconveniences, among others, have encouraged these authors to find an alternative solution that delivers better results in terms of accuracy.

## 3. Analysis of the System

The aim of the proposed system is to increase the accuracy of people positioning inside a room. To do that, let us consider a scenario like that depicted in Figure 1, which represents the generic framework of the system. One or several persons are assumed to be freely moving around a rectangular working area, carrying their own portable device. Each device receives



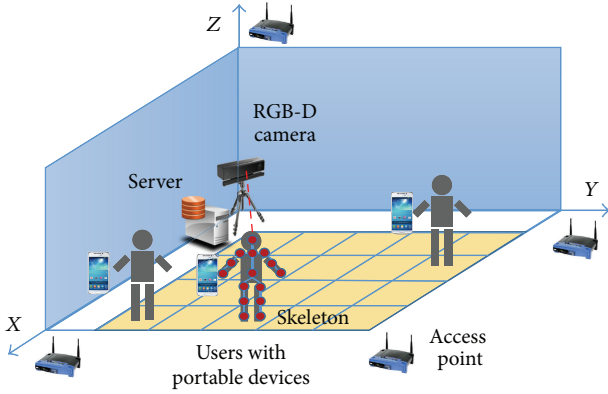


FIGURE 1: Components of the system.

its corresponding wireless signal from one or more APs strategically located in the working area. One RGB-D sensor is placed in such a way that most of the working area is covered. This device delivers a *depth map* and a color image of the scene that are used to identify the 3D skeletons of users. The skeletons are obtained by means of the techniques presented in [23, 24], where authors propose new algorithms to quickly and accurately predict 3D positions of body joints from depth images. Those methods form a core component of the Kinect gaming platform. From these skeletons, neck coordinates are extracted aiming to position people in the environment. This part of the body is chosen because it is less prone to be occluded by elements in the scenario. Finally, a server computer is used for controlling the overall process.

**3.1. System Working Description.** The developed system is divided into two main stages: *learning* and *running*.

**3.1.1. Learning Stage.** The main purpose of this stage is to create, for the selected working area, a new database with the processed information coming from the two technologies: WPS and RGB-D. During this stage, the *fingerprint map* associated with one user is created by registering simultaneously the RSSI values obtained by their portable device and the coordinates of their skeleton. The user moves alone around the room in order to match each RSSI scan with each skeleton position. This task is performed in three different steps: *WiFi Scan*, *RGB-D Scan*, and *Save data*.

During the *WiFi Scan*, the portable device obtains RSSI values for each AP and sends them to the server. When RSSI values are received, the *RGB-D Scan* is started. This process returns the skeleton of the person detected in the room. The system automatically saves the RSSI data and, additionally, the user coordinates of the skeleton are obtained from the *depth map*. Figure 2 shows the system diagram.

To simplify the positioning process without significant loss of precision, other essential tasks are carried out at the end of this stage: The floor of the working area is divided into rectangular cells and RSSI data are grouped in each cell, using the cell position of the skeleton.

The division in cells is produced when the maximum and minimum coordinates of  $X$  and  $Y$  (see Figure 1) have

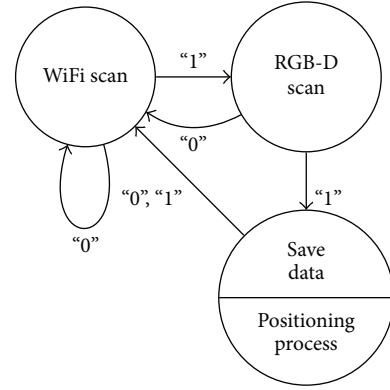


FIGURE 2: System diagram during learning/running stage where values 1 and 0 represent successful or unsuccessful state execution, respectively. “Save data” is the third step during learning stage while “positioning process” is the step during running stage.

been obtained. The coordinates of the skeleton  $(x, y)$  deliver the cell where the user is located  $(c_x, c_y)$ , according to the following:

$$c_x = n_x \cdot \frac{x - \min_x}{\max_x - \min_x}, \quad (1)$$

$$c_y = n_y \cdot \frac{y - \min_y}{\max_y - \min_y},$$

where  $n_x$  and  $n_y$  represent the number of cells in each axis while the variables  $\max_x$ ,  $\min_x$ ,  $\max_y$ , and  $\min_y$  represent the highest and lowest values of each axis (obtained from the *depth map*). Note that  $Z$  coordinates are not considered as the user position is estimated in 2D.

An RSSI vector is created for each cell, pairing each component to the centroid for all of the RSSI measurements for a certain AP (see Figure 3). This allows reducing RSSI variability.

**3.1.2. Running Stage.** This stage represents the normal way of working of the system. It is performed by using the three different steps shown in Figure 2 and considers that several users are moving around the room.

While the *WiFi Scan* is running, each user synchronously sends its RSSI values to the web server. When these data are received, the *RGB-D Scan* starts aiming to obtain the skeletons of people detected in the room. Finally, the *positioning process* estimates the position of each user by combining both data sets in such a way that each skeleton is linked to each RSSI scan.

In the *positioning process*, different algorithms are executed depending on the system’s running mode, going from the simplest *Basic Mode*, in which only WPS method is applied, to more sophisticated ones, where both types of sensors are combined so that each skeleton is linked to each RSSI scan.

The system stores the different RSSI measurements received from the *WiFi Scan* in a table (see Table 1) and the skeleton coordinates obtained from the *RGB-D Scan* in a

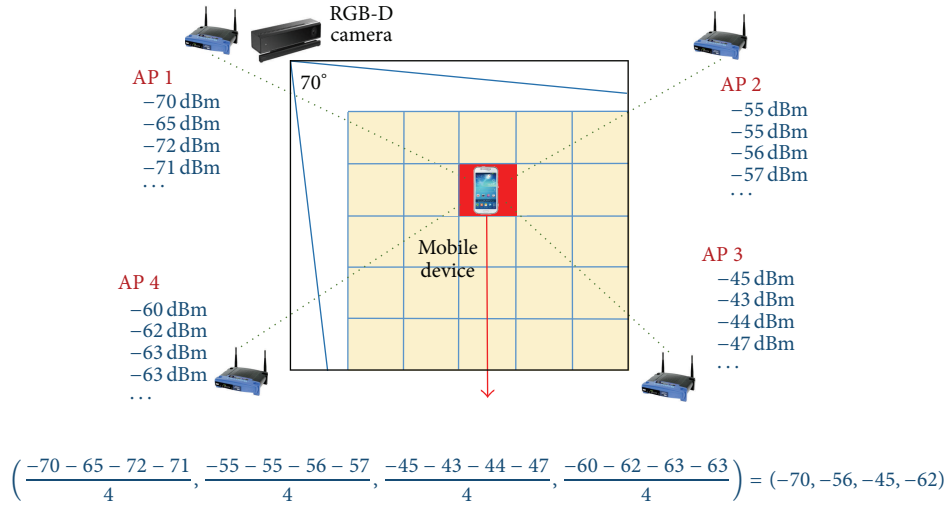


FIGURE 3: Obtaining RSSI vector for one cell (example with 4 APs).

TABLE 1: Example of RSSI table.

Time stamp	User	BSSID	SSID	RSSI
1 (8:31:57)	A	dc:53:7c:25:2c:36	Router 1	-53
1 (8:31:57)	A	84:9c:a6:fe:0e:34	Router 2	-46
1 (8:31:57)	A	4c:54:99:df:9e:ec	Router 3	-93
1 (8:31:57)	B	dc:53:7c:25:2c:36	Router 1	-74
1 (8:31:57)	B	84:9c:a6:fe:0e:34	Router 2	-40
1 (8:31:57)	B	4c:54:99:df:9e:ec	Router 3	-73
2 (8:31:58)	A	dc:53:7c:25:2c:36	Router 1	-52
2 (8:31:58)	A	84:9c:a6:fe:0e:34	Router 2	-51
2 (8:31:58)	A	4c:54:99:df:9e:ec	Router 3	-89
2 (8:31:58)	B	dc:53:7c:25:2c:36	Router 1	-73
2 (8:31:58)	B	84:9c:a6:fe:0e:34	Router 2	-41
2 (8:31:58)	B	4c:54:99:df:9e:ec	Router 3	-72

TABLE 2: Example of skeleton coordinates (neck coordinates).

Time stamp	User	X	Y	Z
1 (8:31:57)	M	-0,298601	0,035828	1,237208
2 (8:31:58)	N	0,229597	-0,025246	1,738968

different one (Table 2). Note that users A and B in Table 1 are not related to users M and N in Table 2. During the *positioning process*, the system will be able to decide if A corresponds to M or N and, in the same manner, if B corresponds to M or N. Skeleton coordinates and RSSI data are linked by a time stamp.

The structure of the recorded RSSI data contains the Basic Service Set Identifier (BSSID), the Service Set Identifier (SSID), RSSI, and the time stamp. BSSID is formed by the Media Access Control (MAC) of each AP. SSID corresponds to the name used by the APs.

BSSID is used instead of SSID. SSID is informative and can be repeated in WLAN since different APs may have the same network name. RSSI data, SSID, and BSSID are collected by the portable devices using the 802.11 layer. At the same time, the portable devices must establish a connection to some accessible network. This can be a WiFi network or a wireless data network of telephony (3G, 4G, etc.). The devices send data, via SOAP protocol through the application layer, to a web server. This web server must be connected to the RGB-D camera but it might not be in the same network as the devices since the web services are available on the Internet.

Different RSSI data entries, as well as different skeletons data entries, can be synchronously produced at the same time stamp, as can be observed in Tables 1 and 2.

As mentioned before, three different running modes are considered in this work: *Basic Mode*, *Improved Mode without Trajectory*, and *Improved Mode with Trajectory*. Their respective features are discussed in next paragraphs.

**Basic Mode: WPS Only.** In this mode, RSSI measurements are obtained from portable devices and compared to the values stored in the *fingerprint* database. During the *learning stage*, the RSSI values of the *fingerprint* were grouped using the centroid of the cells, which reduces RSSI variability.

An error, based on the Euclidean distance between the measured RSSI vector and the RSSI vectors of the centroid of each cell, is calculated. The estimated WPS cell is the one with the lowest error. Equation (2) shows how this error is obtained from two RSSI vectors: the first one read by the portable device and the second one corresponding to the centroid of each cell. Each vector has  $n$  components corresponding to each AP.  $v_{p,new}$  represents the component of the vector for an AP  $p$  where the user is located, while  $v_{p,c_x,c_y}$  represents the component of the centroid vector for that AP  $p$  in each cell  $(c_x, c_y)$ :

$$e_{c_x,c_y} = \frac{1}{n} \cdot \sqrt{\sum_{p \in \text{AP set}} (v_{p,new} - v_{p,c_x,c_y})^2}. \quad (2)$$

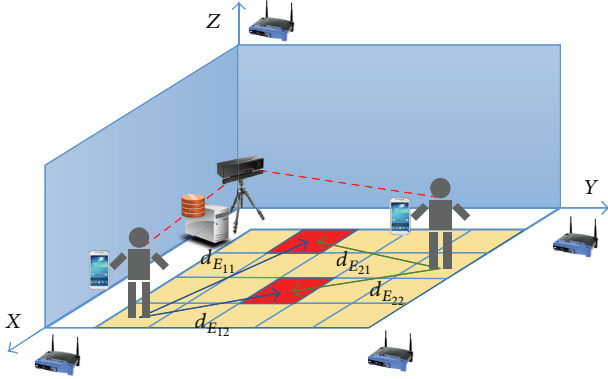


FIGURE 4: Scheme of two people in the room.

*Improved Mode without Trajectory: Combining WPS with Depth Maps.* In this mode, the information provided by *depth maps* helps determine in which cell the user is located. Furthermore, it is useful for clarifying their exact position. The combination of both methods improves indoor positioning in a simple manner.

Two different cases are studied depending on the number of users inside the room: if there is only one user in the room, the *depth map* allows for obtaining the exact position. The portable device provides the right identification of the user.

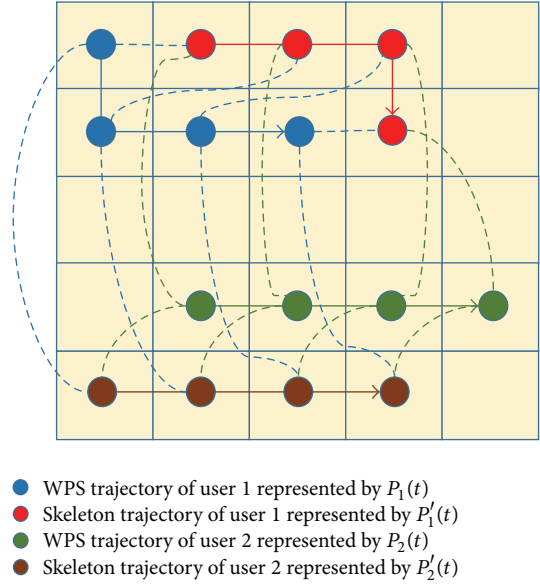
When there are two or more users, as shown in Figure 4, several skeletons are obtained. At the same time, each user sends a group of RSSI measurements to the server. The system initially does not know what skeleton is linked to any particular RSSI data. The proposed method calculates the Euclidean distance between the RSSI data, sent by the smartphones (named WPS cell), and the RSSI centroid of the cell where each skeleton is. Then, the system looks for the best combination between each skeleton and the RSSI measures obtained from each smartphone. This occurs when the sum of the Euclidean distances reaches the minimum:

$$\min \sum_{i,j} s_{ij} \cdot d_{E_{ij}} \wedge s_{ij} \in \{0, 1\} \wedge \sum_x s_{xj} = 1 \wedge \sum_y s_{iy} = 1, \quad (3)$$

where  $s_{ij}$  represents the links between each skeleton and its WPS cell and can take the values 0 and 1.  $d_{E_{ij}}$  represents the Euclidean distance between the skeleton  $i$  and the position  $j$ , where one user has been detected according to WPS.

*Improved Mode with Trajectory: Considering the Trajectory of the User with WPS and Depth Maps.* In this mode, the combination of *depth maps* and WPS also allows for obtaining two different trajectories. The trajectory of the user with WPS represents the cells that the user has previously visited, according to data from WPS. The trajectory of the user in the *depth map* is a group of skeletons obtained for each time stamp. Both trajectories (WPS and skeletons) are synchronized with their time stamps, so when skeletons are received, RSSI values are obtained for all users.

When two or more users are simultaneously in the room and each one has a different trajectory of WPS and skeleton, the system initially does not know what skeleton is linked

FIGURE 5: Example of trajectories of  $n = 2$  users in  $m = 4$  time stamps, including Euclidean distances between the combination of  $P_i(t)$  and  $P'_j(t)$ .

to each user. However, it can calculate it according to an extension of expression (3), as explained in the following.

As mentioned in [25], *synchronized Euclidean distance* measures the distance between two points at identical time stamps. If two trajectories with different points are obtained at the same time (for each pair), the total error is measured as the sum of the distances between all points (points in WPS trajectory and points in skeleton trajectory) at synchronized time stamps.

Figure 5 shows the WPS and skeleton trajectories of 2 users at 4 time stamps.  $P_i(t)$  represents the WPS position of the user  $i$  at the time stamp  $t$ .  $P'_j(t)$  represents the skeleton position of the user  $j$  at the time stamp  $t$ . Although Figure 5 represents the trajectory of user 1 ( $P_1(t), P'_1(t)$ ) and the trajectory of user 2 ( $P_2(t), P'_2(t)$ ), the trajectories of each user are not linked. The system initially does not know if  $P_1(t)$  is associated with  $P'_1(t)$  or  $P'_2(t)$  and in the same way if  $P_2(t)$  is associated with  $P'_1(t)$  or  $P'_2(t)$ . To solve this problem, Expression (4) is used:

$$\begin{aligned} \min \quad & \sum_{i,j} s_{ij} \cdot \sum_{t=1}^m d_E(P_i(t), P'_j(t)) \wedge s_{ij} \in \{0, 1\} \wedge \sum_x s_{xj} \\ & = 1 \wedge \sum_y s_{iy} \\ & = 1. \end{aligned} \quad (4)$$

This expression takes into account the *synchronized Euclidean distance* computing the sum of distances between each pair of points (WPS and skeleton trajectory) and looking for the best combination between WPS trajectories and skeleton trajectories to obtain the minimum sum of all distances of all trajectories. Figure 5 shows all of the different

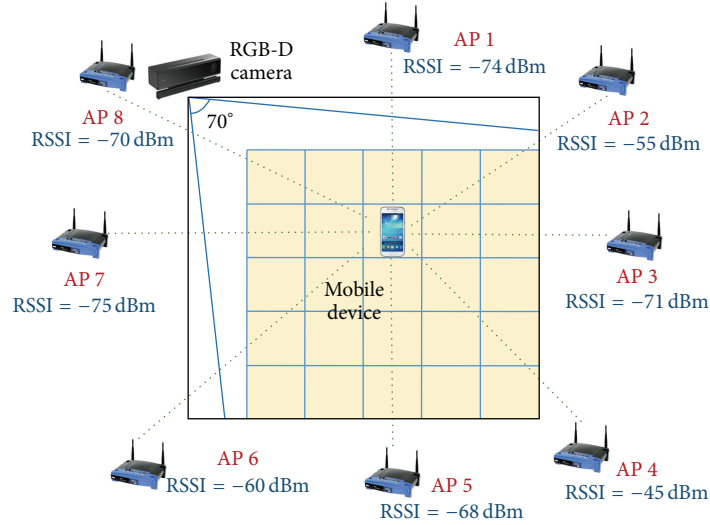


FIGURE 6: Scheme of the room where the tests have been performed.

synchronized Euclidean distances that are computed for 4 time stamps, where there are two users with two different skeleton trajectories and two different WPS trajectories, respectively.

#### 4. Experimentation and Results

Several experiments have been developed in a  $4.5 \times 4.5$  meters room where various APs are available (see Figure 6). An RGB-D sensor was located in one of the corners. In this case, eight APs have been located at different positions of the building. Four of them were situated at less than 6 meters from the user. Various users have participated in the process using portable devices, *smartphones* running Android.

One RGB-D sensor based on time-of-flight technology (ToF), Kinect v2, has been employed in these experiments. This device delivers up to 2 MPx images ( $1920 \times 1080$ ) at 30 Hz and 0.2 MPx *depth maps* with a resolution of  $512 \times 424$  pixels. This Kinect camera is connected to a web server where data is saved and processed.

The horizontal field of view of the RGB-D sensor is  $70^\circ$  so, as shown in Figure 6, it is only able to detect people in a section of the room (in yellow). This section has a size of  $3.71 \times 3.71$  meters.

During the *learning stage*, one user has generated the *fingerprint map* and the matching with the skeletons. The user has moved around the room to produce 1000 different measurements. They have been obtained periodically, sending the RSSI values to a web service hosted on the server. Each time this web service was called, a skeleton scan was performed and the coordinates of the neck were saved, aiming to represent the position of the user.

At the end of the *learning stage*, the floor has been divided into 25 cells ( $5 \times 5$  square cells of 0.74 meters side) and the RSSI centroids for each cell have been calculated. RSSI scans have been grouped according to the distance between their original associated skeleton and the center of each cell.

TABLE 3: Results of different experiments.

Use of trajectory	Successes (2 users)	Successes (3 users)
No	73%	46%
Yes	89%	71%

**4.1. Positioning Experiments.** The results obtained in *Basic Mode* show that the WPS error of positioning a person inside a room is higher than 2 meters. This high error does not allow for distinguishing between different users. For this reason, different experiments have been carried out, with one, two, and three users simultaneously to prove the efficiency of the *Improved Mode*.

In the case of one user, the positioning succeeded in 100% of cases because RGB-D sensor detects just one skeleton. When there are several users simultaneously, RSSI values are synchronously sent to a positioning web service at the same time stamp. The server obtains a skeleton capture of all users present in the room and finally calculates and returns their positions. One result is satisfactory when the system is able to correctly detect the cells where the users are. If each user is situated in a different cell, the system also determines their right positions according to the skeletons.

250 tests have been done, considering 2 or 3 users in the room. The results show that when trajectory is not used, two users are properly detected in 73% of cases and three users in 46% of cases. Most of the errors are produced when the users are in the same cell. When the trajectories of the users are taken into account, the results improve considerably. As shown in Table 3, a comparison of the results has been done at four different time stamps.

The results show that the best performance is obtained when the users are initially in different cells. When the number of users increases, the performance is lower because there are more skeleton trajectories for each WPS trajectory. But considering that the results are above 71% for three users



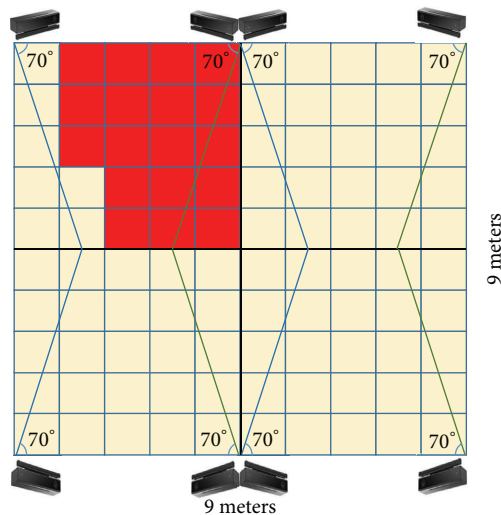


FIGURE 7: System configuration in larger environments.

in a small room (4.5 meters  $\times$  4.5 meters), the system delivers an accurate position of the users in most of the cases.

## 5. Conclusions

This work presents a new method for indoor positioning based on the combination of WPS with *fingerprinting* and the use of *depth maps*. One RGB-D sensor has been used to obtain the *depth maps* and, subsequently, the skeletons. The combination of both technologies is a simple and economical system that increases the performance of WPS in interiors. The accuracy of WPS detecting users in cells of  $2 \times 2$  meters in a room is lower than 50%. The proposed method allows improving the results until reaching more than 89% for two users and 70% for three.

The combination of WPS and *depth maps* presents some advantages such as low cost, the use of simple devices (i.e., *smartphones*), and easy installation. Furthermore, the system is not intrusive since the identity of users is not required.

The method proposed is open to use *crowdsensing* [4], because it is possible to add knowledge without doing new learning. If there is just one user in the environment, the system would be able to recalculate the RSSI centroids for each cell, using the new data obtained from the user (RSSI values and skeleton). This technique would adjust the parameters continuously during system operation.

Besides the number of users, the system is scalable to bigger environments. However, the Kinect sensor has a limited range of a few meters. For this reason, it would be necessary to use more than one device. Figure 7 shows a configuration for a room of 9 meters side. RGB-D sensors are placed aiming to cover the wider angle possible. In this manner, eight cameras would scan the whole room. This figure shows in red the area that would be covered by the top-left sensor. Some sensors cover an overlapped area, which would improve the system accuracy.

Other non-low-cost commercial devices allow obtaining depth maps in wider ranges. For example, Peregrine 3D Flash

LIDAR Vision System [26] is a lightweight camera able to capture a depth map in 5 nanoseconds with the help of a Class I laser. It can operate with lenses of  $60^\circ$  and a range over 1 Km.

Despite the fact that this work just estimates the current position of the users, it would be possible to predict their forthcoming path by means of their last trajectories, considering the simultaneous evaluation of WPS and skeleton trajectories.

## Competing Interests

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

## Acknowledgments

This work has been developed with the help of the Research Project DPI2013-44776-R of MICINN. It also belongs to the activities carried out within the framework of the research network CAM RoboCity2030 S2013/MIT-2748 of Comunidad de Madrid.

## References

- [1] G. Deak, K. Curran, and J. Condell, "A survey of active and passive indoor localisation systems," *Computer Communications*, vol. 35, no. 16, pp. 1939–1954, 2012.
- [2] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, 2007.
- [3] M. R. U. Saputra, Widyawan, G. D. Putra, and P. I. Santosa, "Indoor human tracking application using multiple depth-cameras," in *Proceedings of the 4th International Conference on Advanced Computer Science and Information Systems (ICACSIS '12)*, pp. 307–312, December 2012.
- [4] K. Subbu, C. Zhang, J. Luo, and A. Vasilakos, "Analysis and status quo of smartphone-based indoor localization systems," *IEEE Wireless Communications*, vol. 21, no. 4, pp. 106–112, 2014.
- [5] W. Liu, Y. Chen, Y. Xiong, L. Sun, and H. Zhu, "Optimization of sampling cell size for fingerprint positioning," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 273801, 6 pages, 2014.
- [6] M. Quan, E. Navarro, and B. Peuker, "Wi-Fi localization using RSSI fingerprinting," Tech. Rep., California Polytechnic State University, 2010.
- [7] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 2, pp. 775–784, March 2000.
- [8] X. Feng, Z. Gao, M. Yang, and S. Xiong, "Fuzzy distance measuring based on RSSI in Wireless Sensor Network," in *Proceedings of 3rd International Conference on Intelligent System and Knowledge Engineering (ISKE '08)*, vol. 1, pp. 395–400, November 2008.
- [9] J. Svejčko, M. Malajner, and D. Gleich, "Distance estimation using RSSI and particle filter," *ISA Transactions*, vol. 55, pp. 275–285, 2015.

- [10] E. Martin, O. Vinyals, G. Friedland, and R. Bajcsy, "Precise indoor localization using smart phones," in *Proceedings of the 18th ACM International Conference on Multimedia ACM Multimedia (MM '10)*, pp. 787–790, Firenze, Italy, October 2010.
- [11] C. Kornuta, N. Acosta, and J. M. Toloza, "Posicionamiento WIFI con variaciones de fingerprint," in *Proceedings of the 18th Congreso Argentino de Ciencias de la Computación*, 2013.
- [12] G. Deak, K. Curran, and J. Condell, "Filters for RSSI-based measurements in a device free passive localisation scenario," *Image Processing & Communications*, vol. 15, pp. 23–34, 2010.
- [13] F. Evennou and F. Marx, "Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, Article ID 86706, 11 pages, 2006.
- [14] M. N. Husen and S. Lee, "Indoor human localization with orientation using WiFi fingerprinting," in *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication (ICUIMC '14)*, article 109, ACM, Siem Reap, Cambodia, January 2014.
- [15] G. Ye, Y. Liu, Y. Deng et al., "Free-viewpoint video of human actors using multiple handheld kinects," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1370–1382, 2013.
- [16] H. Takizawa, S. Yamaguchi, M. Aoyagi, N. Ezaki, and S. Mizuno, "Kinect cane: object recognition aids for the visually impaired," in *Proceedings of the 6th International Conference on Human System Interactions (HSI '13)*, pp. 473–478, IEEE, Sopot, Poland, June 2013.
- [17] Y. Nakano, K. Izutsu, K. Tajitsu, K. Kai, and T. Tatsumi, "Kinect Positioning System (KPS) and its potential applications," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation*, November 2012.
- [18] C. K. Schindhelm, "Evaluating slam approaches for microsoft kinect," in *Proceedings of the 18th International Conference on Wireless and Mobile Communications (ICWMC '12)*, pp. 402–407, Venice, Italy, 2012.
- [19] P. Mirowski, R. Palaniappan, and T. K. Ho, "Depth camera SLAM on a low-cost WiFi mapping robot," in *Proceedings of the IEEE International Conference on Technologies for Practical Robot Applications (TePRA '12)*, pp. 1–6, Woburn, Mass, USA, April 2012.
- [20] J. Biswas and M. Veloso, "Multi-sensor mobile robot localization for diverse environments," in *RoboCup 2013: Robot World Cup XVII*, pp. 468–479, Springer, 2014.
- [21] C.-S. Wang and C.-L. Chen, "RFID-based and Kinect-based indoor positioning system," in *Proceedings of the 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE '14)*, pp. 1–4, Aalborg, Denmark, May 2014.
- [22] S. Y. Lee, B. C. Min, D. H. Kim, and J. S. Yoon, "Passive RFID positioning system using RF power control," in *Robot Intelligence Technology and Applications*, pp. 845–853, Springer, New York, NY, USA, 2013.
- [23] J. Shotton, T. Sharp, A. Kipman et al., "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.
- [24] A. Barmpoutis, "Tensor body: real-time reconstruction of the human body and avatar synthesis from RGB-D," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1347–1356, 2013.
- [25] C. T. Lawson, S. Ravi, and J. H. Hwang, "Compression and mining of GPS trace data: new techniques and applications," Tech. Rep., Region II University Transportation Research Center, 2011.
- [26] Peregrine 3D Flash LIDAR Vision System, Advanced Scientific Concepts, <http://www.advancedscientificconcepts.com/products/Peregrine.html>.

## Research Article

# Cable Crosstalk Suppression in Resistive Sensor Array with 2-Wire S-NSDE-EP Method

JianFeng Wu<sup>1</sup> and Lei Wang<sup>2</sup>

<sup>1</sup>*Jiangsu Key Lab of Remote Measurement and Control, School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China*

<sup>2</sup>*School of Automation, Nanjing Institute of Technology, Nanjing 211167, China*

Correspondence should be addressed to JianFeng Wu; [wjf@seu.edu.cn](mailto:wjf@seu.edu.cn)

Received 8 December 2015; Revised 26 January 2016; Accepted 28 January 2016

Academic Editor: Fernando Torres

Copyright © 2016 J. Wu and L. 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.

With long flexible cables connected to the 1-wire setting non-scanned-driving-electrode equipotential (S-NSDE-EP) circuit, the resistive sensor array modules got flexibility in robotic operations but suffered from the crosstalk problem caused by wire resistances and contacted resistances of the cables. Firstly, we designed a new S-NSDE-EP circuit using two wires for every driving-electrode and every sampling-electrode to reduce the crosstalk caused by the connected cables in the 2D networked resistive sensor array. Then, an equivalent resistance expression of the element being tested (EBT) for this circuit was analytically derived. Then, the 1-wire S-NSDE-EP circuit and the 2-wire S-NSDE-EP circuit were evaluated by simulations. The simulation results show that the 2-wire S-NSDE-EP circuit, though it requires a large number of wires, can greatly reduce the crosstalk error caused by wire resistances and contacted resistances of the cables in the 2D networked resistive sensor array.

## 1. Introduction

Resistive sensor arrays were widely used in tactile sensing [1–8], light sensing [9], infrared sensing [10], and so forth. In robotic applications, long flexible cables were preferred for flexibility and limited space of the sensitive areas. With tested cables of lengths from 55 mm to 500 mm (as shown in Table 1), different modules of resistive sensor arrays were connected to the test circuits through the plugs and the sockets. Vidal-Verdú et al. [1, 3] designed and compared circuits of networked piezoresistive sensor arrays. Speeter [2] designed a flexible sensing system with  $16 \times 16$  resistive taxels. Yang et al. [4] designed a  $32 \times 32$  flexible array within a  $160 \text{ mm} \times 160 \text{ mm}$  temperature and tactile sensing area. Zhang et al. [5] reported a  $3 \times 3$  thin tactile force sensor array based on conductive rubber. Castellanos-Ramos et al. [6] reported a  $16 \times 16$  tactile sensor array based on conductive polymers with screen-printing technology. Kim et al. [7] reported a flexible tactile sensor array with high performance in sensing contact force. Lazzarini et al. [8] reported a  $16 \times 16$  tactile sensor array for practical applications in manipulation.

But cables had different wire resistances which increased with the increase of their lengths. Between the plugs of the connected cables and the sockets of the test circuits, there existed contacted resistances of tens of milliohms to several ohms varying with the variation of mechanical vibration and time. But new methods are still lacking, which can be used to suppress crosstalk caused by long cables.

For this purpose, we present a novel cable crosstalk suppression circuit based on a 2-wire method for the 2D networked resistive sensor arrays in the row-column fashion. This paper begins with an overview of the application fields of the 2D networked resistive sensor arrays. Secondly, a novel cable crosstalk suppression method will be proposed and its equivalent resistance expression of the element being tested (EBT) will be analytically derived. Then simulations will be implemented to evaluate this method with different parameters such as wire resistances and contacted resistances of the cables, the array size, the measurement range of the EBT, and the adjacent elements' resistances of 2D networked resistive sensor arrays. Finally, the results of experiments will be analyzed and conclusions for the method will be given.

TABLE 1: Resistive sensor arrays with cables of different lengths.

Literature	Sensor	Array size of sensing elements	Cable length (mm)	Cable crosstalk
[1]	Polymer based FSR	$16 \times 9$	$>55$	Yes
[2]	FSR	$16 \times 16$	$>60$	Yes
[3]	FSR	$16 \times 16$	$>70$	Yes
[4]	Conductive rubber	$32 \times 32$	$>70$	Yes
[5]	Conductive rubber	$3 \times 3$	$>95$	Yes
[6]	Conductive polymer	$16 \times 16$	$>100$	Yes
[7]	Semiconductor strain gage	$5 \times 5$	$>100$	Yes
[8]	FSR	$16 \times 16$	500	Yes
[9]	Light dependent resistor	$16 \times 16$	—	Yes

## 2. Principle Analyses

In the row-column fashion, 2D resistive sensor arrays needed few wires but suffered from crosstalk caused by parasitic parallel paths. For suppressing crosstalk, many methods have been proposed and analyzed in literatures, such as the passive integrators method [3], the inserting diode method [11], the resistive matrix array method [12], the voltage feedback methods [2, 13–17], and the zero potential methods (ZPMs) [1, 3–10, 16–20]. Wu et al. have suppressed the crosstalk caused by the adjacent column elements and the adjacent row elements with the Improved Isolated Drive Feedback Circuit (IIDFC) [13] and the Improved Isolated Drive Feedback Circuit with Compensation (IIDFCC) [14]. Wu et al. have also proposed a general voltage feedback circuit model [15] for fast analyzing the performances of different voltage feedback circuits. D'Alessio has analyzed measurement errors in the scanning circuits of piezoresistive sensors arrays [16]. Saxena et al. [18, 19] have suppressed the crosstalk caused by the adjacent column elements with large number of op-amps using the zero potential method. Roohollah et al. [20] have suppressed the crosstalk error caused by the input offset voltage and input bias current of the op-amp with a novel double-sampling technique. In these methods, the measurement accuracy of the EBT still suffered from cable crosstalk.

Liu et al. [17] defined the setting non-scanned-electrode zero potential (S-NSE-ZP) method, the setting non-scanned-sampling-electrode zero potential (S-NSSE-ZP) method, and the setting non-scanned-driving-electrode zero potential (S-NSDE-ZP) method for the zero potential methods, in which bipolar power sources were necessary for op-amps and analog digital converters (ADCs). In some circuits [1, 3], the reference voltages were not zero, so op-amps and ADCs with unipolar power sources, which were of less cost and were more convenient for use, could be used. So we defined those equipotential methods as the setting non-scanned-electrode-equipotential (S-NSE-EP) method, the setting non-scanned-sampling-electrode-equipotential (S-NSSE-EP) method, and the setting non-scanned-driving-electrode-equipotential (S-NSDE-EP) method. In this analysis, the S-NSDE-EP circuit was taken for example. Traditional S-NSDE-EP circuit of resistive networked sensor array in shared row-column fashion was shown as Circuit A in Figure 1(a). In Circuit A, the row electrodes and the column electrodes were used as the

sampling electrodes and the driving electrodes, respectively. In Circuit A,  $R_{11}$  in the  $M \times N$  resistive array was the element being tested (EBT); only one connected wire was used for every column and row electrode between the sensor array and the circuit; only one equal current  $M : 1$  multiplexer was used between the current setting resistor ( $R_{set1}$ ) and the row electrodes of the sensor module. On column electrodes of the circuit,  $2 : 1$  multiplexers had multiplexer switch resistances ( $R_{sc}$ ); column wires had column resistances ( $R_{Lc}$ s) including column wire resistances and column contacted resistances. On row electrodes of the circuit, the equal current  $M : 1$  multiplexers had multiplexer switch resistances ( $R_{sr}$ ); row wires had row resistances ( $R_{Lr}$ s) including row wire resistances and row contacted resistances. Thus Circuit A had one row sampling op-amp, one  $M : 1$  multiplexer,  $N : 2 : 1$  multiplexers, and  $M + N$  wires.

Under an ideal condition, all  $R_{sc}$ s and all  $R_{Lc}$ s were omitted. Thus the voltage ( $V_{cy1}$ ) on the column electrode of the EBT was equal to the feedback voltage ( $V_{xy1}$ ), and the voltages on the non-scanned column electrodes were equal to the reference voltage ( $V_{ref1}$ ). At the same time, all  $R_{sr}$ s and all  $R_{Lr}$ s were omitted. Thus the voltage ( $V_{e1}$ ) on the inverting input of the row sampling op-amp was equal to the voltage ( $V_{rx1}$ ) on the row electrode of EBT. Under the effect of the ideal op-amp,  $V_{e1}$  was equal to  $V_{ref1}$  and the current ( $I_{xy1}$ ) on the EBT was following the change of the current ( $I_{set1}$ ) on  $R_{set1}$ . As the voltages on the non-scanned column electrodes were equal to  $V_{ref1}$ , the currents on the adjacent row elements of EBT were equal to zero. At the same time, the current on the inverting input of the ideal op-amp was omitted for its infinite input impedance, the current ( $I_{xy1}$ ) on the EBT was equal to the current ( $I_{set1} = (V_{xy1} - V_{ref1})/R_{xy1} = V_{ref1}/R_{set1}$ ) on  $R_{set1}$ . Thus,  $I_{set1}$  and  $I_{xy1}$  were equal. As  $V_{ref1}$  and  $R_{set1}$  were known,  $V_{xy1}$  could be measured by ADC, so the equivalent resistance value ( $R_{xy1}$ ) of the EBT in Circuit A could be calculated with the following:

$$R_{xy1} = \frac{(V_{xy1} - V_{ref1}) \times R_{set1}}{V_{ref1}}. \quad (1)$$

But under the real condition as shown in Figure 1(b),  $V_{cy1}$  was not equal to  $V_{xy1}$  for  $R_{sc}$  and  $R_{Lc}$ , and  $V_{e1}$  was not equal to  $V_{rx1}$  for  $R_{sr}$  and  $R_{Lr}$ . The ideal feedback condition was destroyed by the row wires and the column wires, so extra measurement errors of the EBT existed.



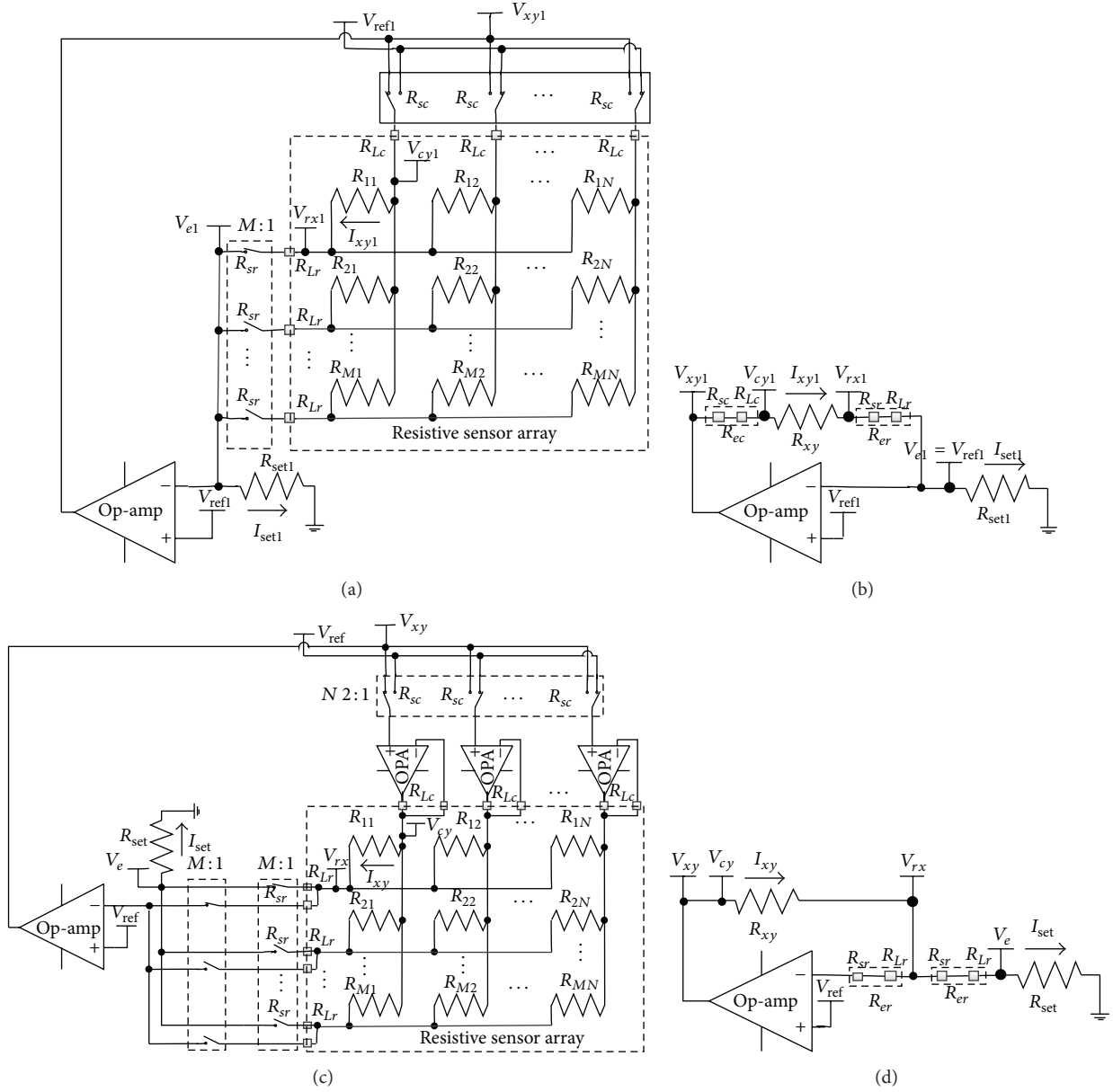


FIGURE 1: (a) 1-wire S-NSDE-EP circuit (Circuit A). (b) Simplified measurement circuit of 1-wire S-NSDE-EP circuit (Circuit B). (c) 2-wire S-NSDE-EP circuit (Circuit C). (d) Simplified measurement circuit of 2-wire S-NSDE-EP circuit (Circuit D).

For suppression cable crosstalk in the 2D networked resistive arrays, we proposed a 2-wire equipotential method (Circuit C, as shown in Figure 1(c)). In Circuit C, we used two wires for every row electrode and every column electrode between the sensor module and the test circuit; also we used one column driving op-amp for every column electrode and one more equipotential  $M:1$  multiplexer between the row electrodes and the row sampling op-amp. Thus Circuit C had one row sampling op-amp,  $N$  column driving op-amps,  $N:2:1$  multiplexers, two  $M:1$  multiplexers, and  $2(M+N)$  connected wires.

Every column electrode in the sensor module was connected with the output of its column driving op-amp by one driving wire and it was also connected with the inverting

input of its column driving op-amp by one driving sampling wire. The noninverting input of every column driving op-amp was connected with the common port of its column  $2:1$  multiplexer; thus every noninverting input was connected with  $V_{xy}$  or  $V_{ref}$ . The noninverting input of EBT's column driving op-amp was connected with  $V_{xy}$  and the noninverting inputs of other column driving op-amps were connected with  $V_{ref}$ .

As the input impedance of every column driving op-amp was much bigger than  $R_{sc}$ , the effect of  $R_{sc}$  could be omitted. So the voltage on the noninverting input of every column driving op-amp was equal to the input voltage ( $V_{xy}$  or  $V_{ref}$ ) of its  $2:1$  multiplexer. If the column driving op-amps had sufficient driving ability, the voltage on every column

electrode was following the change of the voltage on the noninverting input of its column driving op-amp. So  $V_{cy}$  was equal to  $V_{xy}$ , and the voltages on nonscanned column electrodes were equal to  $V_{ref}$ . Thus the crosstalk effect of  $R_{Lc}$  and  $R_{sc}$  was suppressed.

By one equal current wire, every row electrode in the sensor module was connected with one channel of the equal current  $M:1$  multiplexer with its common port connected with  $R_{set}$ . In the equal current  $M:1$  multiplexer, only the row electrode of EBT was gated and all other nonscanned electrodes were suspended. So only the row electrode of the EBT was connected with  $R_{set}$ .

By one equipotential wire, every row electrode in the sensor module was also connected with one channel of the equipotential  $M:1$  multiplexer with its common port connected with the inverting input of the row sampling op-amp. In the equipotential  $M:1$  multiplexer, only the row electrode of EBT was gated and all other nonscanned electrodes were suspended. So only the EBT's row electrode was connected with the inverting input of the row sampling op-amp. From the output port of the EBT's column driving op-amp, the test current firstly flowed through the EBT, then it flowed through the row equal current wire, then it flowed through the equal current  $M:1$  multiplexer, and finally it flowed through  $R_{set}$  to ground.

As the input impedance of the row sampling op-amp was much bigger than its series resistances such as the switch resistance of the equipotential  $M:1$  multiplexer, the wire resistance of the equipotential wire, and the contacted resistance, the voltage on the inverting input of the row sampling op-amp was equal to the voltage ( $V_{rx}$ ) of the EBT's row electrode.

Under the effect of the row sampling op-amp, the current ( $I_{xy}$ ) on the EBT followed the change of the current ( $I_{set}$ ) on  $R_{set}$ . As the input impedance of the row sampling op-amp was much bigger than its parallel resistances such as  $R_s$ ,  $R_{sr}$ , and  $R_{Lr}$ , the leak current on the inverting input of the voltage feedback op-amp could be omitted. And the voltage on every nonscanned column electrode was equal to  $V_{ref}$ , which was also equal to  $V_{rx}$ . Thus the currents on the EBT's ( $N-1$ ) row adjacent elements were zero. So  $I_{set}$  was equal to  $I_{xy}$ . The current with equal value also flowed through  $R_{sr}$  and  $R_{Lr}$ . As  $R_{set}$  was known and  $I_{set}$  was equal to  $I_{xy}$ , we could know  $I_{xy}$  if the voltage ( $V_e$ ) on  $R_{set}$  and the voltage ( $V_{xy}$ ) on the EBT were known. Thus we could get  $R_{xy}$  of the EBT.

But  $V_e$  was not equal to  $V_{rx}$  for  $R_{er}$  (as shown in Figure 1(d)) which was the crosstalk caused by the row wire. Thus extra measurement error of the EBT was caused by it. From the above discussion, we could know that the currents on  $R_{xy}$ ,  $R_{set}$ , and  $R_{er}$  had equal values. So we could use (2) to calculate  $R_{xy}$  in Circuit C. We found that  $R_{er}$  did not exist in (2). As  $V_{ref}$  and  $R_s$  were known,  $V_{xy}$  and  $V_e$  could be measured by ADC, so the equivalent resistance value ( $R_{xy}$ ) of the EBT in Circuit C could be calculated with (2). Thus the crosstalk caused by the row wire was suppressed:

$$R_{xy} = \frac{(V_{xy} - V_{ref}) \times R_{set}}{V_e}. \quad (2)$$

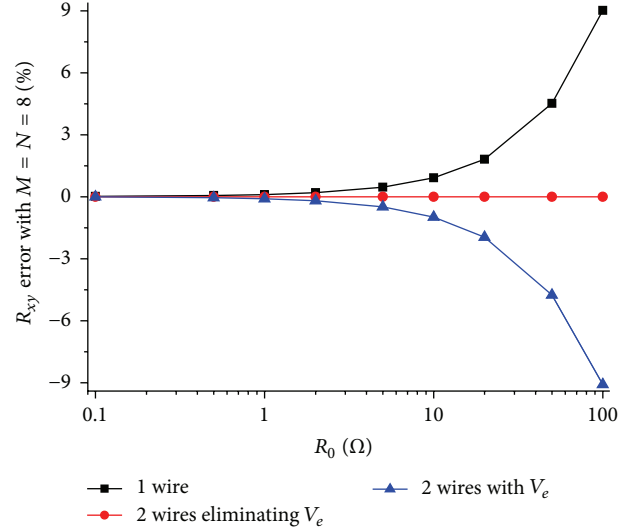


FIGURE 2: Effect of  $R_0$  on the  $R_{xy}$  errors in the 1-wire S-NSDE-EP circuit and the 2-wire S-NSDE-EP circuit where  $M = N = 8$ .

From the above discussion, the 2-wire S-NSDE-EP method can depress the crosstalk caused by the row wires and the column wires such as  $R_{sr}$ s,  $R_{Lr}$ s,  $R_{sc}$ s, and  $R_{Lc}$ s.

### 3. Simulation Experiments and Discussion

To emulate the performance of our method, OP07 was selected as the macromodel of the op-amp (from the datasheet, the offset voltage, the bias current, the gain-bandwidth, and the gain are equal to 75  $\mu$ V, 2.8 nA, 0.60 MHz, and 126 dB, resp.) in the simulations of National Instrument (NI) Multisim 12. In simulations,  $V_{ref}$  was set at 0.1 V,  $R_{set}$  was set at 1 k $\Omega$ , the positive voltage source of the op-amps was set at 9 V, and the negative voltage source of the op-amps was set at -6 V.

**3.1.  $R_0$  Effect Simulation in NI Multisim.** Cable resistance ( $R_0$ ,  $R_0 = R_{er} = R_{ec}$ ) including the wire resistance and the contacted resistance affected the performance of the 2D networked resistive circuits. We investigated the effect of  $R_0$  including wire resistance and contacted resistance on the 1-wire S-NSDE-EP circuit and the 2-wire S-NSDE-EP circuit in NI Multisim. In simulations, we fixed some parameters including all elements in the resistive sensor array at 10 k $\Omega$  and  $M$  and  $N$  at 8, and  $R_0 = R_{er} = R_{ec}$  in sensor arrays varied synchronously with the same resistance value in 0.1  $\Omega$ –100  $\Omega$ . The simulation results of the two circuits in NI Multisim 12 were shown in Figure 2. In the results, as shown in Figure 2, the deviation effect of  $V_e$  caused by the row line and the row multiplexer was also considered.

From Figure 2, with  $R_0$  varied from 0.1  $\Omega$  to 100  $\Omega$ ,  $R_{xy}$  errors in the 1-wire S-NSDE-EP circuit showed a significant change (from 0.025% to 9.017%) with an obvious positive increase coefficient, while  $R_{xy}$  errors in the 2-wire S-NSDE-EP circuit eliminating the deviation effect of  $V_e$  showed a tiny change (from -0.000% to -0.003%). But if the deviation

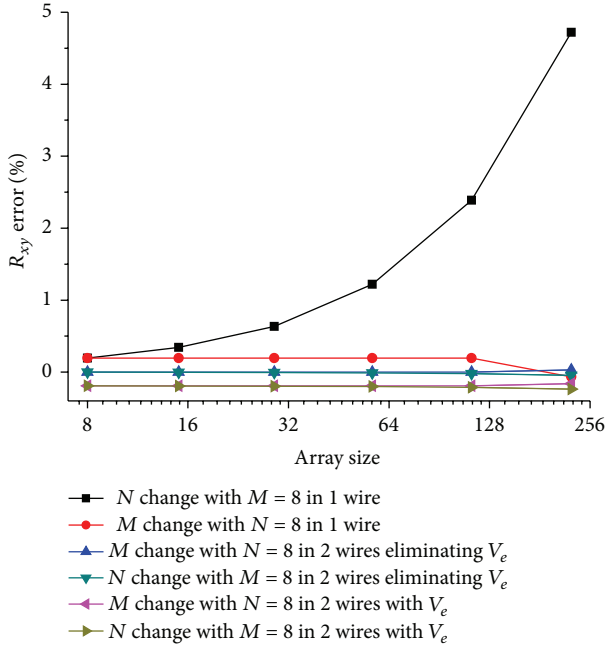


FIGURE 3: Array size effect on the  $R_{xy}$  errors in the 1-wire S-NSDE-EP circuit and the 2-wire S-NSDE-EP circuit where  $R_0 = 2 \Omega$  and  $R_{other} = 10 \text{ k}\Omega$ .

effect of  $V_e$  was ignored,  $R_{xy}$  errors in the 2-wire S-NSDE-EP circuit with  $V_e$  would be significant (from  $-0.002\%$  to  $-9.083\%$ ) as shown in Figure 2. Thus, the 2-wire S-NSDE-EP circuit eliminating the deviation effect of  $V_e$  has a better performance than the 1-wire S-NSDE-EP circuit when  $R_0$  is varied from  $0.1 \Omega$  to  $100 \Omega$ ; the absolute  $R_{xy}$  errors of the 2-wire S-NSDE-EP circuit eliminating the deviation effect of  $V_e$  are small enough to be negligible when  $R_0$  is less than  $100 \Omega$ .

In the data of the simulation results, we also found the offset value of  $V_e$  from  $V_{ref}$  was varied from  $0.19 \text{ mV}$  to  $9.08 \text{ mV}$  with  $R_0$  changing from  $2 \Omega$  to  $100 \Omega$ .

**3.2. Array Size Effect Simulation Experiment.** Parameters of the array size such as the row number ( $M$ ) and the column number ( $N$ ) were proved to have effect on the performance of the 2D networked resistive sensor arrays [9–19]. We investigated the effect of  $M$  and  $N$  on the 1-wire S-NSDE-EP circuit and the 2-wire S-NSDE-EP circuit in NI Multisim. In simulations, we fixed some parameters including all elements in the resistive sensor array at  $10 \text{ k}\Omega$ ,  $M$  or  $N$  at 8, and  $R_0$  at  $2 \Omega$ , and  $N$  or  $M$  was one number in (8, 15, 29, 57, 113, and 225). The results of the array size effect on the 1-wire S-NSDE-EP circuit and the 2-wire S-NSDE-EP circuit were simulated in NI Multisim and the results were shown in Figure 3. In the results, as shown in Figure 3, the deviation effect of  $V_e$  caused by the row line and the row multiplexer was also considered.

From Figure 3, with the increase of the column number, the  $R_{xy}$  errors in the 1-wire S-NSDE-EP circuit had a positive coefficient (from  $0.196\%$  to  $4.722\%$ ) while the  $R_{xy}$  errors in the 2-wire S-NSDE-EP circuit eliminating the deviation effect of  $V_e$  had a negative coefficient (from  $-0.000\%$  to  $-0.044\%$ ).

But if the deviation effect of  $V_e$  was ignored, we found a deviation of  $R_{xy}$  errors (from  $-0.191\%$  to  $-0.235\%$ ) in the 2-wire S-NSDE-EP circuit with  $V_e$  in Figure 3. The absolute  $R_{xy}$  errors in the 2-wire S-NSDE-EP circuit eliminating the deviation effect of  $V_e$  had been reduced significantly comparing with the absolute  $R_{xy}$  errors in the 1-wire S-NSDE-EP circuit.

From Figure 3, with the row number changed in the range from 8 to 113, the  $R_{xy}$  errors in both circuits changed little (from  $0.196\%$  to  $0.194\%$  for the 1-wire S-NSDE-EP circuit, about  $0.000\%$  for the 2-wire S-NSDE-EP circuit eliminating the deviation effect of  $V_e$ , about  $-0.191\%$  for the 2-wire S-NSDE-EP circuit with  $V_e$ ); but when the row number changed in the range from 113 to 225, the  $R_{xy}$  errors in both circuits changed clearly (from  $0.194\%$  to  $-0.067\%$  for the 1-wire S-NSDE-EP circuit, from  $0.000\%$  to  $0.032\%$  for the 2-wire S-NSDE-EP circuit eliminating the deviation effect of  $V_e$ , from  $-0.191\%$  to  $-0.159\%$  for the 2-wire S-NSDE-EP circuit with  $V_e$ ). If every column driving op-amp had a sufficient current driving ability, the row number had less influence on the  $R_{xy}$  errors in both circuits. In the data of the simulation results, we also found the offset value of  $V_e$  from  $V_{ref}$  was about  $0.19 \text{ mV}$  with array size changed.

Thus, in the 2-wire S-NSDE-EP circuit eliminating the deviation effect of  $V_e$ , the influence of array size on the  $R_{xy}$  error has been decreased greatly.

**3.3. The Adjacent Elements Effect Simulation.** In literatures [9–19], the adjacent elements played a significant role in affecting the measurement accuracy of the EBT. In simulations, we fixed some parameters including the resistance value of nonadjacent elements and all other adjacent elements at  $10 \text{ k}\Omega$ ,  $M$  and  $N$  at 8, and  $R_0$  at  $2 \Omega$ . The resistance value of an adjacent element varied in the range from  $0.1 \text{ k}\Omega$  to  $1 \text{ M}\Omega$ . The adjacent element could be an adjacent row element ( $R_{adjr}$ ) or an adjacent column element ( $R_{adjc}$ ). The simulation results of the 1-wire S-NSDE-EP circuit and the 2-wire S-NSDE-EP circuit in NI Multisim were shown in Figures 4–7.

From Figures 4–7, the  $R_{xy}$  errors of the EBT of both circuits had negative coefficient when the resistance value of the EBT increased; the  $R_{xy}$  errors of the EBT showed irregular variations when the resistances of the EBT was bigger than a certain value ( $\geq 30 \text{ k}\Omega$  for the 1-wire S-NSDE-EP circuit,  $\geq 50 \text{ k}\Omega$  for the 2-wire S-NSDE-EP circuit). We found that the output voltages of the row sampling op-amp in both circuits were saturated for a bigger resistance value of the EBT. Under the same power source voltage, the measurement range of the 2-wire S-NSDE-EP circuit was bigger than that of the 1-wire S-NSDE-EP circuit.

From Figures 4–7, the  $R_{xy}$  errors of the EBT with a bigger resistance value were susceptible to interference from by one  $R_{adjr}$  or one  $R_{adjc}$  with a smaller resistance value. In both circuits, the changes of the  $R_{xy}$  errors for the change of one  $R_{adjr}$  were bigger than the changes of the  $R_{xy}$  errors for the change of one  $R_{adjc}$ . With one  $R_{adjr}$  or one  $R_{adjc}$  varied from  $0.1 \text{ k}\Omega$  to  $1 \text{ M}\Omega$ , the changes of the  $R_{xy}$  errors (with  $R_{xy}$  at  $30 \text{ k}\Omega$ , from  $-0.307\%$  to  $-0.048\%$  for one  $R_{adjc}$  and from  $-3.022\%$  to  $-0.051\%$  for one  $R_{adjr}$ ) in the 1-wire S-NSDE-EP circuit were significant, while those (with  $R_{xy}$  at  $50 \text{ k}\Omega$ ,

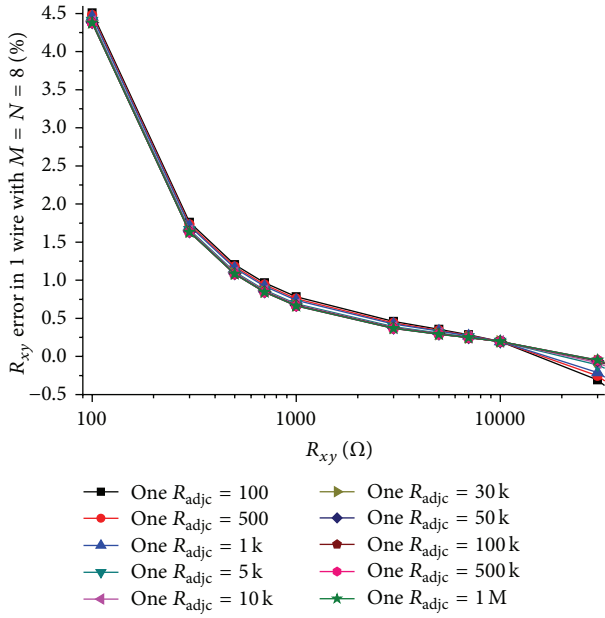


FIGURE 4: The  $R_{adj}$  effect on  $R_{xy}$  errors in the 1-wire S-NSDE-EP circuit.

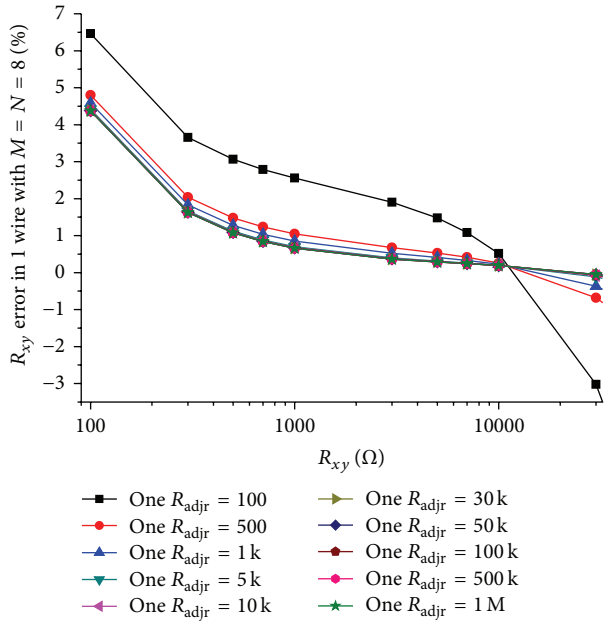


FIGURE 5: The  $R_{adj}$  effect on  $R_{xy}$  errors in the 1-wire S-NSDE-EP circuit.

from  $-0.006\%$  to  $-0.006\%$  for one  $R_{adj}$  and from  $-0.106\%$  to  $-0.005\%$  for one  $R_{adj}$  in 2-wire S-NSDE-EP circuit were small. Thus, in the 2-wire S-NSDE-EP circuit, the influence of the adjacent elements on the  $R_{xy}$  error has been decreased greatly.

**3.4. The Op-Amp's Offset Voltage Effect Simulation.** As many op-amps were used in the 2-wire S-NSDE-EP circuit, the offset voltages of the op-amps would affect the performance

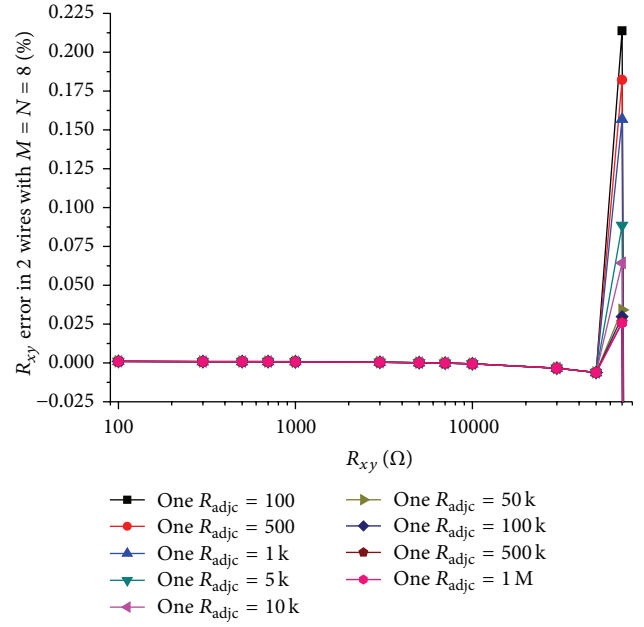


FIGURE 6: The  $R_{adj}$  effect on  $R_{xy}$  errors in the 2-wire S-NSDE-EP circuit.

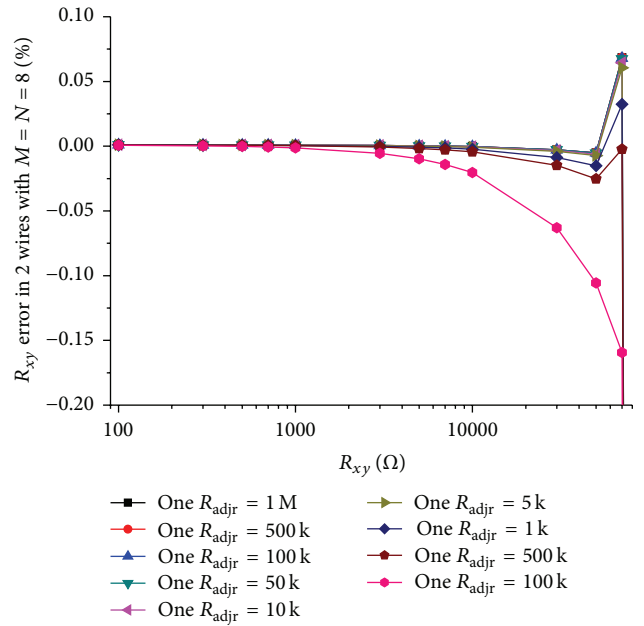


FIGURE 7: The  $R_{adj}$  effect on  $R_{xy}$  errors in the 2-wire S-NSDE-EP circuit.

of the proposed circuit. In simulations, we fixed some parameters including the resistance value of all other row elements at  $10\text{ k}\Omega$ ,  $M$  and  $N$  at 8,  $R_0$  at  $2\text{ }\Omega$ , and all  $R_{ajcr}$ s at the same resistance value in ( $100\text{ }\Omega$ ,  $300\text{ }\Omega$ ,  $1\text{ k}\Omega$ , and  $10\text{ k}\Omega$ ). The offset voltages of the nonscanned column driving op-amps varied synchronously with the same value in ( $-75\text{ }\mu\text{V}$ – $75\text{ }\mu\text{V}$ ), and the 2-wire S-NSDE-EP circuit was simulated in NI Multisim and the results were shown in Figure 8.



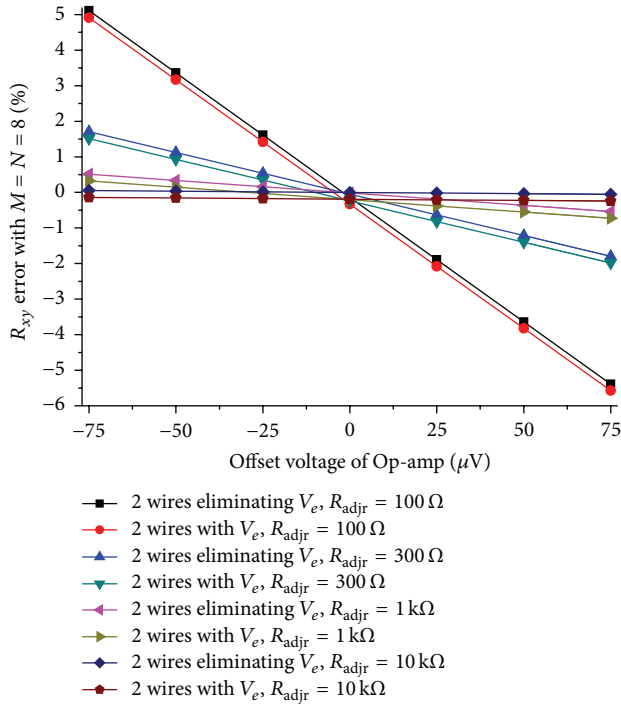


FIGURE 8: The op-amp's offset voltage effect on  $R_{xy}$  errors in the 2-wire S-NSDE-EP circuit.

From Figure 8, we found that the offset voltages of the op-amps and the resistances of the row adjacent elements affected the 2-wire S-NSDE-EP circuit. The smaller these resistances were and the larger the offset voltage was, the larger the  $R_{xy}$  error in the proposed circuit was.

**3.5. The Op-Amp's Driving Capability Effect Simulation.** The op-amp's driving capability affected the performance of the 2-wire S-NSDE-EP circuit. The nonscanned elements' bypass effect on the EBT in the 2D resistive sensor array was obvious when the EBT had large resistance and all nonscanned elements had the small resistances. In the worst case, the EBT had the maximum resistance and all nonscanned elements had the minimum resistances [17]. In the experiments, we were about to simulate the op-amp's driving capability with all nonscanned elements of different fixed small resistances and the EBT of a large resistance. In simulations, we fixed some parameters including  $M$  and  $N$  at 8 and  $R_0$  at  $2 \Omega$  and all non-scanned elements at the same resistance value in ( $100 \Omega$ ,  $300 \Omega$ ,  $500 \Omega$ ,  $1 \text{ k}\Omega$ , and  $3 \text{ k}\Omega$ ). The resistance value of the EBT varied in the range from  $0.1 \text{ k}\Omega$  to  $60 \text{ k}\Omega$ . The 2-wire S-NSDE-EP circuit with the op-amp of OP07 was simulated in NI Multisim and the results were shown in Figure 9 and Table 2. Also the op-amps of OP07 ( $I_{\text{short-circuit}} = 30 \text{ mA}$ ) were replaced by the op-amps of AD797 ( $I_{\text{driving}} = 50 \text{ mA}$ ), and the 2-wire S-NSDE-EP circuit was simulated.

From Figure 9 and Table 2, with the resistances of all non-scanned elements fixed, the 2-wire S-NSDE-EP circuit failed to work normally when the EBT's resistance exceeded certain values; with the minimum resistances of all nonscanned elements increased, the maximum resistance which could

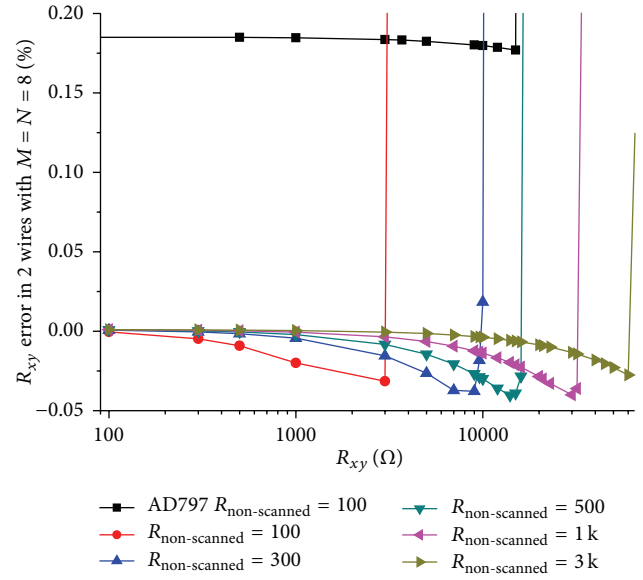


FIGURE 9: The op-amp's driving capability effect on  $R_{xy}$  errors in the 2-wire S-NSDE-EP circuit.

TABLE 2: The EBT's maximum resistance of the 2-wire S-NSDE-EP circuit with its op-amp of OP07.

$R_{\text{non-scanned}}$ (k $\Omega$ )	The maximum resistance (k $\Omega$ )
0.10	3.2
0.30	9.6
0.50	16.0
1.00	32.0
3.00	60.0

be tested in the 2-wire S-NSDE-EP circuit increased; with a larger op-amp's driving capability, the 2-wire S-NSDE-EP circuit with its op-amp of AD797 had a larger measurement range.

**3.6. Discussion.** From the results in Figure 1, the 1-wire S-NSDE-EP circuit had one voltage feedback op-amp,  $N : 1$  multiplexers, one  $M : 1$  multiplexers, and  $M + N$  wires; the 2-wire S-NSDE-EP circuit had one voltage feedback op-amp,  $N$  column driving op-amps,  $N : 1$  multiplexers, two  $M : 1$  multiplexers, and  $2(M + N)$  wires. Thus more components and more wires were used in the 2-wire S-NSDE-EP circuit.

From the results in Figure 2, the 2-wire S-NSDE-EP method was verified to be efficient in depressing the crosstalk caused by the row wires and the column wires such as  $R_{sf}$ ,  $R_{Lr}$ ,  $R_{sc}$ , and  $R_{Lc}$ . It should be noticed that all conductions were right under the assumption that the column driving op-amps had sufficient driving ability and the row sampling op-amp had very big input impedance on its inverting input.

From the results in Figures 3, 6, and 7, the 2-wire equipotential circuit was failed to work normally with too much big resistance value of the EBT. If the resistance of the adjacent elements in resistive sensor array was too small, the absolute  $R_{xy}$  errors of the EBT would increase significantly. At the same time, if the row sampling op-amp did not have

very big input impedance or the elements in resistive sensor array had very big resistance values for the row sampling op-amp's input impedance,  $I_{xy}$  would be not equal to  $I_{set}$ . Thus the ideal work conditions were destroyed for the 2-wire S-NSDE-EP circuit and the  $R_{xy}$  error would be significant.

From the results in Figures 2, 3, and 8,  $V_e$  in the 2-wire S-NSDE-EP circuit had a significant effect on the  $R_{xy}$  error when the resistances such as the wire resistance, the contacted resistance, and the switch-on resistance of the equal current  $M:1$  multiplexer were large. Thus the deviation value of  $V_e$ , mainly caused by the connected cable and the equal current  $M:1$  multiplexers should be carefully considered in the 2-wire S-NSDE-EP circuit. In the proposed method, the deviation effect of  $V_e$  had been eliminated and the 2-wire S-NSDE-EP circuit with good performance was obtained. As the offset value of  $V_e$  from  $V_{ref}$  was varied from 0.19 mV to 9.08 mV with  $R_0$  changing from  $2\ \Omega$  to  $100\ \Omega$ , one more op-amp was necessary for amplifying the signal of  $V_e$  in the case of using an analog-digital converter with limited resolution in the 2-wire S-NSDE-EP circuit.

From the results in Figure 8, the offset voltages of the column driving op-amps had an obvious influence on the performance of the 2-wire S-NSDE-EP circuit, and the offset voltage's effect would be more obvious for the element being tested with its row adjacent elements of smaller resistance values. With the increase of the offset voltage, the  $R_{xy}$  error increased. As the column number of the sensor array had accumulation influence on the conductance values of the row adjacent elements, it would enhance the effect of the offset voltage. Obviously, the offset voltage of the row sampling op-amp had similar influence on the performance of the 2-wire S-NSDE-EP circuit. Thus in the practical circuit, the op-amps with smaller offset voltages were preferred. In the op-amp's offset voltage effect simulation experiments, the offset voltages of all nonscanned driving op-amps varied synchronously with the same value and their effect was obvious. But, in a practical circuit, the op-amps' offset voltages would be the uncertain values less than the offset voltage given in their datasheets and their effect would be weaker. In the 2-wire S-NSDE-EP circuit, the double-sampling technique [20] was also useful for eliminating the effect of those nonidealities of the op-amps such as the input offset voltage and the input bias current.

From the results in Figure 9 and Table 2, the op-amp's driving capability affected the measurement range of the 2-wire S-NSDE-EP circuit; with the op-amp fixed, there was an approximate linear relation between the minimum resistance and the maximum resistance in the 2-wire S-NSDE-EP circuit. But the maximum resistance which could be tested in the 2-wire S-NSDE-EP circuit was also limited by the test current and the power source voltage. Thus the op-amps with large driving capability were preferred in the 2-wire S-NSDE-EP circuit. But the op-amps with large driving capability always had a large offset voltage. So the contradiction between the driving capability affecting its measurement range and the offset voltage affecting its measurement accuracy should be balanced according to the test requirement.

For good performance of the IIDFC [13] and the IIDFCC [14], special compensated resistors with their resistances

equal to their multiplexers' switch-on resistances are necessary. But the multiplexers' switch-on resistances may vary in the practical circuits, and the ideal performances of the IIDFC and the IIDFCC are difficult to realize. In the 2-wire S-NSDE-EP method, two wires for every row electrode and every column electrode between the sensor module and the test circuit, though it requires a large number of wires, are easier to achieve. The 2-wire S-NSDE-EP method's performance and its limitation have been verified by simulation experiments. Similar methods can also be used in the S-NSSE-EP circuit and the S-NSE-EP circuit. But these should be verified in future practical application.

## 4. Conclusion

Firstly, a 2-wire S-NSDE-EP method of the 2D networked resistive sensor array was proposed. Secondly, the formula was given for the equivalent resistance expression of the element being tested in the networked sensor array by principle analyses. Then, the effects of some parameters on the measurement accuracy of the EBT were simulated with the National Instrument Multisim 12, the parameters including the wire resistances and the contacted resistances of long cables, the array size and the adjacent elements of the 2D resistive sensor array, and the offset voltages of the op-amps. The simulation results show that the 2-wire equipotential method was verified to be efficient in depressing the crosstalk caused by the row wires and the column wires such as  $R_{sr}$ ,  $R_{Lr}$ ,  $R_{sc}$ , and  $R_{Lc}$ ; in the 2D networked resistive sensor array with the 2-wire S-NSDE-EP circuit, the influence of the adjacent column elements and the adjacent row elements on the measurement error of the element being tested has been reduced greatly. Finally, the factors which affected the performance of the 2-wire S-NSDE-EP circuit were discussed and the conclusion was given.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgment

This study was supported by the Specialized Research Fund Program for the Doctoral Program of Higher Education (no. 20130092110060). This study was also supported by the Scientific Research Fund Project of Nanjing Institute of Technology (no. CKJB201405), the Open Fund of the Key Laboratory of Remote Measurement and Control Technology in Jiangsu Province (nos. YCCK201401 and YCCK201006), the National Major Scientific Equipment R&D Project (Grant no. ZDYZ2010-2), and NSAF (no. U1230114).

## References

- [1] F. Vidal-Verdú, M. Jose Barquero, J. Castellanos-Ramos et al., "A large area tactile sensor patch based on commercial force sensors," *Sensors*, vol. 11, no. 5, pp. 5489–5507, 2011.

- [2] T. H. Speeter, "A tactile sensing system for robotic manipulation," *The International Journal of Robotics Research*, vol. 9, no. 6, pp. 25–36, 1990.
- [3] F. Vidal-Verdú, Ó. Oballe-Peinado, J. A. Sánchez-Durán, J. Castellanos-Ramos, and R. Navas-González, "Three realizations and comparison of hardware for piezoresistive tactile sensors," *Sensors*, vol. 11, no. 3, pp. 3249–3266, 2011.
- [4] Y.-J. Yang, M.-Y. Cheng, S.-C. Shih et al., "A  $32 \times 32$  temperature and tactile sensing array using PI-copper films," *The International Journal of Advanced Manufacturing Technology*, vol. 46, no. 9, pp. 945–956, 2010.
- [5] X. Zhang, Y. Zhao, and X. Zhang, "Design and fabrication of a thin and soft tactile force sensor array based on conductive rubber," *Sensor Review*, vol. 32, no. 4, pp. 273–279, 2012.
- [6] J. Castellanos-Ramos, R. Navas-González, H. Macicior, T. Sikora, E. Ochoteco, and F. Vidal-Verdú, "Tactile sensors based on conductive polymers," *Microsystem Technologies*, vol. 16, no. 5, pp. 765–776, 2010.
- [7] M.-S. Kim, H.-J. Shin, and Y.-K. Park, "Design concept of high-performance flexible tactile sensors with a robust structure," *International Journal of Precision Engineering and Manufacturing*, vol. 13, no. 11, pp. 1941–1947, 2012.
- [8] R. Lazzarini, R. Magni, and P. Dario, "A tactile array sensor layered in an artificial skin," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction and Cooperative Robots*, vol. 3, pp. 114–119, Pittsburgh, Pa, USA, August 1995.
- [9] R. S. Saxena, R. K. Bhan, and A. Aggrawal, "A new discrete circuit for readout of resistive sensor arrays," *Sensors and Actuators A: Physical*, vol. 149, no. 1, pp. 93–99, 2009.
- [10] R. S. Saxena, R. K. Bhan, C. R. Jalwania, and S. K. Lomash, "A novel test structure for process control monitor for un-cooled bolometer area array detector technology," *Journal of Semiconductor Technology and Science*, vol. 6, no. 4, pp. 299–312, 2006.
- [11] D. Prutchi and M. Arcan, "Dynamic contact stress analysis using a compliant sensor array," *Measurement*, vol. 11, no. 3, pp. 197–210, 1993.
- [12] L. Shu, X. Tao, and D. D. Feng, "A new approach for readout of resistive sensor arrays for wearable electronic applications," *IEEE Sensors Journal*, vol. 15, no. 1, pp. 442–452, 2015.
- [13] J. F. Wu, L. Wang, and J. Q. Li, "Design and crosstalk error analysis of the circuit for the 2-D networked resistive sensor array," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 1020–1026, 2015.
- [14] J. F. Wu, L. Wang, J. Q. Li, and A. G. Song, "A novel crosstalk suppression method of the 2-D networked resistive sensor array," *Sensors*, vol. 14, no. 7, pp. 12816–12827, 2014.
- [15] J. F. Wu, L. Wang, and J. Q. Li, "General voltage feedback circuit model in the two-dimensional networked resistive sensor array," *Journal of Sensors*, vol. 2015, Article ID 913828, 8 pages, 2015.
- [16] T. D'Alessio, "Measurement errors in the scanning of piezoresistive sensors arrays," *Sensors and Actuators A: Physical*, vol. 72, no. 1, pp. 71–76, 1999.
- [17] H. Liu, Y.-F. Zhang, Y.-W. Liu, and M.-H. Jin, "Measurement errors in the scanning of resistive sensor arrays," *Sensors and Actuators A: Physical*, vol. 163, no. 1, pp. 198–204, 2010.
- [18] R. S. Saxena, R. K. Bhan, N. K. Saini, and R. Muralidharan, "Virtual ground technique for crosstalk suppression in networked resistive sensors," *IEEE Sensors Journal*, vol. 11, no. 2, pp. 432–433, 2011.
- [19] R. S. Saxena, S. K. Semwal, P. S. Rana, and R. K. Bhan, "Crosstalk suppression in networked resistive sensor arrays using virtual ground technique," *International Journal of Electronics*, vol. 100, no. 11, pp. 1579–1591, 2013.
- [20] Y. Roohollah, A. Safarpour, and R. Lotfi, "An improved-accuracy approach for readout of large-array resistive sensors," *IEEE Sensor Journal*, vol. 16, no. 1, pp. 210–215, 2015.