UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

LAURO JOSÉ LYRIO JÚNIOR

IMAGE-BASED MAPPING AND LOCALIZATION USING VG-RAM WEIGHTLESS NEURAL NETWORKS

VITÓRIA
2014

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

LAURO JOSÉ LYRIO JÚNIOR

MAPEAMENTO E LOCALIZAÇÃO BASEADOS EM IMAGEM UTILIZANDO REDES NEURAIS SEM PESO DO TIPO VG-RAM

VITÓRIA
2014

LAURO JOSÉ LYRIO JÚNIOR

IMAGE-BASED MAPPING AND LOCALIZATION USING VG-RAM WEIGHTLESS NEURAL NETWORKS

Dissertation submitted to the Graduate Program in Computer Science from the Technology Center of the Federal University of Espírito Santo, as a partial requirement for obtaining the degree of Master of Science in Computer Science.

VITÓRIA
2014

LAURO JOSÉ LYRIO JÚNIOR

MAPEAMENTO E LOCALIZAÇÃO BASEADOS EM IMAGEM UTILIZANDO REDES NEURAIS SEM PESO DO TIPO VG-RAM

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

VITÓRIA
2014

LAURO JOSÉ LYRIO JÚNIOR

IMAGE-BASED MAPPING AND LOCALIZATION USING VG-RAM WNN

COMISSÃO EXAMINADORA

Prof. Dr. Alberto Ferreira De Souza

Universidade Federal do Espírito Santo

Orientador

Prof. Dr. Thiago Oliveira Dos Santos

Universidade Federal do Espírito Santo

Coorientador

Profa. Dra. Claudine Badue

Universidade Federal do Espírito Santo

Profa. Dra. Marley Maria B. R. Vellasco

Pontifícia Universidade Católica do Rio de Janeiro

Vitória, 25 de Agosto de 2014.

Abstract

Mapping and localization are fundamental problems in autonomous robotics. Autonomous robots need to know where they are in their operational area to navigate through it and to perform activities of interest. In this work, we present an image-based mapping and localization system that employs Virtual Generalizing Random Access Memory Weightless Neural Networks (VG-RAM WNN) for localizing an autonomous car.

In our system, a VG-RAM WNN learns world positions associated with images and three-dimensional landmarks captured along a trajectory, in order to build a map of the environment. During the localization, the system uses its previous knowledge and uses an Extended Kalman Filter (EKF) to integrate sensor data over time through consecutive steps of state prediction and correction. The state prediction step is computed by means of our robot's motion model, which uses velocity and steering angle information computed from images using visual odometry. The state correction step is performed by integrating the VG-RAM WNN learned world positions in combination to the matching of landmarks previously stored in the robot's map. Our system efficiently solves the (i) mapping, (ii) global localization and (iii) position tracking problems using only camera images.

We performed experiments with our system using real-world datasets, which were systematically acquired during laps around the *Universidade Federal do Espírito Santo* (UFES) main campus (a 3.57 km long circuit). Our experimental results show that the system is able to learn large maps (several kilometres in length) of real world environments and perform global and position tracking localization with mean pose precision of about 0.2m compared to the Monte Carlo Localization (MCL) approach employed in our autonomous vehicle.

Resumo

Localização e Mapeamento são problemas fundamentais da robótica autônoma. Robôs autônomos necessitam saber onde se encontram em sua área de operação para navegar pelo ambiente e realizar suas atividades de interesse. Neste trabalho, apresentamos um sistema para mapeamento e localização baseado em imagens que emprega Redes Neurais Sem Peso do Tipo VG-RAM (RNSP VG-RAM) para um carro autônomo.

No nosso sistema, uma RNSP VG-RAM aprende posições globais associadas à imagens e marcos tridimensionais capturados ao longo de uma trajetória, e constrói um mapa baseado nessas informações. Durante a localização, o sistema usa um Filtro Estendido de Kalman para integrar dados de sensores e do mapa ao longo do tempo, através de passos consecutivos de predição e correção do estado do sistema. O passo de predição é calculado por meio do modelo de movimento do nosso robô, que utiliza informações de velocidade e ângulo do volante, calculados a partir de imagens utilizando-se odometria visual. O passo de correção é realizado através da integração das posições globais que a RNSP VG-RAM com a correspondência dos marcos tridimensional previamente armazenados no mapa do robô.

Realizamos experimentos com o nosso sistema usando conjuntos de dados do mundo real. Estes conjuntos de dados consistem em dados provenientes de vários sensores de um carro autônomo, que foram sistematicamente adquiridos durante voltas ao redor do campus principal da UFES (um circuito de 3,57 km). Nossos resultados experimentais mostram que nosso sistema é capaz de aprender grandes mapas (vários quilômetros de comprimento) e realizar a localização global e rastreamento de posição de carros autônomos, com uma precisão de 0,2 metros quando comparado à abordagem de Localização de Monte Carlo utilizado no nosso veículo

autônomo.

Acknowledgements

I would like to give a very special thanks to my supervisor Alberto Ferreira De Souza for his friendship, encouragement, attention, support, kindness, and invaluable advice throughout these 2 years.

I would also like to give thanks to Franco Machado from Mogai company for his support and for allowing me to do my master's degree while working in several other research projects.

My thanks to my professors Thiago Oliveira dos Santos, Claudine Badue, Mariella Berger and my friends Filipe Wall Mutz, Lucas de Paula Veronese, Bruno Oliveira, Avelino Forechi and all other friends of the *Departamento de Informática* at UFES for their encouragement, advice, and criticism.

I gratefully acknowledge the financial support of the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) and UFES.

Finally, but by no means the least, I wish to thank my family (specially my mother, Ivanda Maria Pedrini Lyrio), and my girlfriend, Gabriela Gomes Antunes, whose, despite all the problems and difficulties, were always supporting me. A very special thanks to you!

To my family

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Mapping and localization are fundamental problems in autonomous robotics. Autonomous robots need to know where they are in their operational area to navigate through it and to perform activities of interest. Therefore, they need consistent maps of the environment and the ability to localize themselves in these maps using sensor data.

The localization problem can be branched along a number of sub-problems according to the nature of the environment and the initial knowledge that a robot has about its location [THR05]. Considering the type of initial knowledge, we can qualify the localization problem into three different branches: global localization, position tracking and the kidnapped robot problem.

Global localization is the ability to resolve the robot's position in a previously learned map, given no information other than that the robot is around someplace in the map. Once the initial robot's position is found in the map, the position tracking is the problem of keeping track of that position over time. Generally, the global localization problem is harder than position tracking and the kidnapped robot problem is even more difficult than global localization. In the kidnapped robot problem, a well localized robot is moved to an unknown place and it needs to relocalize itself. The solution of the kidnapped robot problem ensures that the robot has the appropriate abilities to

recover from localization failures.

Many probabilistic approaches have been proposed to solve the localization sub-problems mentioned above [THR05, BEE04, DIS01, BUR96]; however, some of these sub-problems are more difficult to solve than others. Global localization, for instance, is more challenging than position tracking, and localization and mapping are currently harder to perform with cameras than with Light Detection and Ranging (LIDAR) systems. Nevertheless, the development of efficient mapping and localization techniques based on cameras is relevant for the widespread use of these techniques, because cameras are much cheaper than laser system and the amount of information (color, depth, resolution) that they provide is relatively higher than that delivered by LIDARs.

In this work, we present a novel image-based mapping and localization approach which employs Virtual Generalizing Random Access Memory (VG-RAM) Weightless Neural Networks (WNN) [LUD99], dubbed VG-RAM Image-Based Mapping and Localization (VIBML) (Figure 1.1)

Figure 1.1: Illustration of VIBML performing global localization and position tracking around the UFES' campus. VIBML uses previously learned image-pose pairs stored in a neural map to estimate global poses (red cars) from currently observed images. VIBML's neural position tracking keeps a smooth trajectory (green dots), even in case of global localization failure (purple car).

The VIBML system efficiently solves the problems of mobile robot mapping, global localization and position tracking using only camera images.

But, although VIBML performs mapping and localization, it does not map the environment and simultaneously localizes the robot (it is not a Simultaneous Mapping and Localization (SLAM) system [THR05, DIS01]).

The VIBML system mimics the human capacities of learning about a place, recognizing a previously learned area and localizing itself while moving through the environment as well. Memorizing images of places and labels associated with them (road names, addresses, etc.) and then, in later moments, to remember the labels when the same images are seen again is a task that humans perform very well. Similarly, in the mapping phase, VIBML receives images of the environment, positions (labels) where images were captured, as well as characteristic points that belong to these images. Subsequently, it learns associations between the images, positions and the images' characteristic points and represent them as a map of the environment (it learns about a place). In the localization phase, VIBML receives images of the environment and uses its previously acquired knowledge – "the map" – to output the positions and the characteristic points representing the places the system believes these images were captured. Finally, it uses those positions and characteristic points to perform global localization (it recognizes a place) and position tracking (it localizes itself while moving through the environment).

We have tested the VIBML system with a set of mapping and localization experiments using real-world datasets. These datasets consist of data from various sensors acquired systematically during laps performed by an autonomous car in a 3.57 km long circuit. These datasets were constructed for this work and are made publicly available with the corresponding ground-truth at www.lcad.inf.ufes.br/log.

Our results shown that our system, purely based on camera images, is capable of localizing robots on large maps (several kilometers in length). Our system was able to localize an autonomous car for a distance of 3.57km around the *Universidade Federal do Espírito Santo* (UFES), with a mean difference of 0.2m when compared to the Occupancy Grid Mapping (OGM)

and Monte Carlo Localization (MCL) solutions [THR05], employed in our autonomous car. In addition, VIBML was able to localize our autonomous car with average positioning error of 1.12m and with 75% of the poses with error below 1.5m.

1.1 Motivation

With the knowledge advancement in the field of probabilistic robotics [THR05, BEE04, DIS01, BUR96], today, is possible to implement an autonomous vehicle, i.e, a passenger vehicle able to drive itself without any assistance, given its starting position and a desired destination [BUE07, DAR11, ROL02, ROU11].

The role played by an autonomous car can be performed by a trained human being (able to drive) without much difficulty thanks to its capabilities of visual cognition, like depth perception, object and edge recognition, colour processing and so on.

We believe that the eyes has a very important role when someone drives a car. In traffic, someone usually makes use of all its human senses to drive. But the vision sense is responsible for much of the work. Detect a sign traffic, localize itself and identify the lane's boundaries are tasks that our brain plays simultaneously using the inputs coming from our eyes [DEL94, JOC95].

The motivation of this work is to better understand the cognitive aspects related to the vision when someone drives a car. In this work we are particularly interested on how human beings learn about a place, and then, with the acquired knowledge, recognizes that place and is able to localize themselves in that, only using the eyes.

1.2 Objectives

The objectives of the present work are to build a system able to mimic the human skill of mapping and localization using mathematical-computational

models inspired in the human biology, and to compare it to the currently used probabilistic approaches for mapping and localization of the literature.

With these goals, we rely on the cognitive aspects of human vision to develop computational models for mapping, global location and position tracking that could be integrated into a platform for autonomous driving.

In order to replace the currently used localization systems, which makes use of sensors as LIDARs (which are very expensive!) for mapping and localization, our system only use cameras, a sensor that is very similar to the human eye, and very cheap.

1.3 Contributions

The main contributions of this work are:

1. Conception of an Image-Based approach that uses VG-RAM WNNs for mapping and localization – the VIBML – capable of localizing robots in GPS-denied condition with a low investment cost, since only cameras are used.
2. Comparison of the VIBML's performance with other probabilistic approaches, specifically, Monte Carlo Localization in Ocuppancy Grid Maps using LIDARs -- like the Velodyne HDL 32-E [THR05].
3. Building of sensor data logs, benefiting the widespread development of algorithms in the field of autonomous robotics.

1.4 Organization of this Dissertation

This dissertation is organized as follows. After this introduction, Chapter 2 presents the related literature for this work. Chapter 3 describes the VIBML system and the mapping, global localization and positioning tracking subsystem in details. Chapter 4 presents the methodology used to carry out the experiments to evaluate the VIBML system and the metrics used in the evaluation. In Chapter 5, we describe the experiments used for investigating

the localization performance of VIBML's global localization and position tracking subsystems. In Chapter 6 we make a discussion and a critical assessment of this work by examining the limitations of our system with respect to robot's pose initialization, the problem of kidnapped robot and the processing time of the subsystems. Chapter 7 presents a summary of this dissertation, its conclusions, and suggests future directions for improving the VIBML system.

Chapter 2

Related Work

Most of the work on robotics vision in the last decade relied on visual features with certain degree of invariance to affine transformations [LOW99, BAY06] (e.g. rotation, translation, scale) for providing robust landmarks for mapping and localization [SSE01, WOL02]. Se et al. [SSE01], for instance, developed a vision-based indoor mobile robot SLAM algorithm using stereo and Scale Invariant Feature Transform (SIFT), while Wolf et al. [WOL02] used invariant features based on image histograms for indoor localization using cameras. Both approaches (and many similar ones) are mainly conceived as map-based indoor localization and may not be suitable for large outdoor environments, as our approach is. In addition, the continuous global localization problem is not solved completely in this works. For instance, in the work of Se, they are not able to perform global localization as VIBML does, because the matching of SIFT features is done only locally.

Several more recent works focus on situations in which only the initial position of the robot is given. In the seminal work of Nister et al., for example [NIS04], visual features present in pairs of consecutive video frames are matched and estimation of the camera motion is computed from the feature tracks. This technique (named visual odometry) is very useful to estimate the motion of a mobile system; however, visual odometry does not keep a map of the environment. Davison et al., in another seminal work [DAV07], developed

a SLAM algorithm that tracks a large set of image features from monocular or stereo video and builds a 3D map of features. Lategahn et al. [LAT11] also track a large set of features from stereo images using the EKF SLAM and compute dense feature maps using them. A similar approach was proposed by Geiger et al. [GEI11], where a sparse feature matcher in conjunction with a visual odometry algorithm were used for generating maps of consistent 3D point-clouds. In spite of their capabilities for visual odometry and/or map construction, none of these techniques is suitable for continuous global localization.

RatSLAM [MIL08] is a biologically inspired SLAM approach that uses a simplified visual odometry in addition to appearance-based image template-matching for building maps consisting of simulated cells activations. The system performance was evaluated on a 66 km long urban street, with many loops. Results showed that RatSLAM is capable of building maps online, close loops and re-localize through sequences of familiar visual scenes, i.e. it is capable of global localization; however, this global localization requires several image frames, while VIBML needs only one image to remember the pose of a previously learned place.

RatSLAM was tested in conjunction with FAB-MAP [GLO10] that is another appearance-based SLAM. FAB-MAP [CUM08] is similar to our work, since it allows continuous global localization by detecting that an image is similar to a previously learned image. However, FAB-MAP is based in the bag-of-words image retrieval systems developed in the computer vision community [SIV03] and its learning algorithm is costly, while VIBML is based on WNN that learns in one shot. In addition, FAB-MAP does not have position tracking functionalities as VIBML has.

SeqSLAM [MIL12] is another state-of-the-art appearance-based SLAM that calculates the best candidate matching of an image within a segment of a sequence of previously seen images. Although this approach can handle normal and extreme conditions in environment appearance even for long running distances, SeqSLAM needs to process a long sequence to recognizes a

previously seen place and it is not able to perform position tracking.

Chapter 3

Image-Based Mapping and Localization with VG-RAM WNN

The VG-RAM Image-Based Mapping and Localization (VIBML) system is composed of three main subsystems: VG-RAM Image-Based Mapping (VIBM), VG-RAM Image-Based Global Localization (VIBGL), and VG-RAM Image-Based Position Tracking (VIBPT).

The VIBM subsystem (bounded by a red rectangle in Figure 3.1) is responsible to create an internal representation of the environment. It firstly receives images of the environment captured by a stereo camera as well as the poses (position and orientation) where these images were captured. Then, it detects characteristics points on the received images, and computes their three dimensional positions (3D landmarks) using distance information from a depth map computed by a stereo matching algorithm. Finally, VIBM learns about the images, the associated poses and landmarks' positions, and constructs the map of the environment, which is represented internally by the contents of the memories of its VG-RAM neurons – the Neural Map.

The VIBGL subsystem (bounded by a blue rectangle in Figure 3.1) is responsible for the system start up and for continuous global localization. It receives images of the environment and uses the previously acquired knowledge – the Neural Map – to output the poses and associated landmarks'

positions where these images were captured.

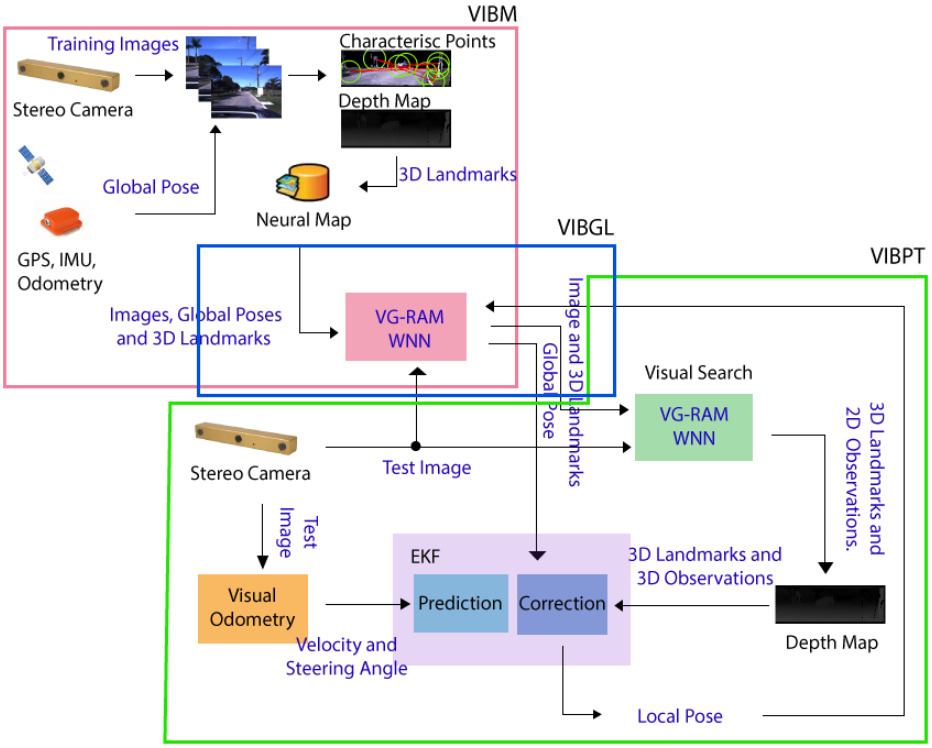


Figure 3.1: The VIBML system architecture. The VIBM subsystem (bounded by a red rectangle) uses images and associated global poses and characteristics points' (or landmarks') positions to build the map of the environment, which is represented internally by the contents of the memories of its neurons. The VIBGL subsystem (bounded by a blue rectangle) uses previously acquired knowledge – the map – to output the global poses where these images were captured. The VIBPT subsystem (bounded by a green rectangle) uses an Extended Kalman Filter (EKF) to integrate sensor data over time through consecutive steps of state prediction and correction. The state prediction step is computed by means of our robot's motion model, which uses odometry information, and the state correction step is performed by integrating the global poses estimated by VIBGL with the matching of landmarks previously stored in the map.

The VIBPT subsystem (bounded by a green rectangle in Figure 3.1) is responsible for keeping track of new poses over times. It employs an Extended Kalman Filter - EKF [THR05, SIM06] to integrate sensor readings over time through consecutive steps of state prediction and correction. The state prediction step is computed by means of our robot's motion model, which uses velocity and steering angle information computed from images using visual odometry [GEI11]. The state correction step is performed in two steps.

In the first step, VIBPT receives an image of the environment and consults VIBGL for the most similar image and respective 3D landmarks in the Neural Map. Subsequently, VIBPT projects the 3D landmarks outputted by

VIBGL back to the camera's coordinate system (2D coordinates of characteristic points) and searches for these characteristic points in the previously received image of the environment, using an approach for visual search based on VG-RAM WNN [SOU13]. Once the correspondences for each characteristic point is found, VIBPT computes their three dimensional positions (3D observations) using the distance information from a depth map computed by a stereo matching algorithm and corrects the robot's local pose by adjusting it in proportion to the difference between the 3D landmarks and the 3D observations using a measurement model.

In the second step, VIBPT adjusts the robot's local pose by fusing the corrected local pose with the global pose estimated by VIBGL, which ensures that the local pose error is bounded by the global pose error.

In the next section we explain in details the basic component of all VIBML's subsystems, the VG-RAM WNN.

3.1 VG-RAM WNN

The VG-RAM WNN is a very effective machine learning technique that offers easy implementation and fast training procedure, thanks to its simplicity [LUD99]. Such neural networks comprise a set of neural layers composed of VG-RAM neurons connected to other layers through synapses.

A basic network architecture comprises two layers: an input layer and a neural layer. Differently from weighted neural networks, that store knowledge in their synapses, in VG-RAM WNNs each neuron of a neural layer has a set of weightless synapses $S = \{s_1, ..., s_p\}$. The data read from the corresponding input layer through the synapses are transformed in a vector of bits $I = \{i_1, ..., ip\}$ (one bit per synapse). Each bit of this vector is computed using a synapse mapping function that transforms non-binary values from the input layer in binary values.

The VG-RAM WNN neurons store knowledge in private local memories that work as look-up tables and keep sets $L = \{L_1, ..., L_j, ..., L_m\}$ of pairs L_j =

(I_j, t_j), where I_j is a binary input vector and t_j is its corresponding output label. The binary input vectors are extracted from the input layers via the set S of synapses of each neuron, while the output labels t are the learned neurons' output-values for each binary input vector I.

The VG-RAM WNN supervised training and test work as follows. During training, an input pattern and its expected output pattern are set in the input layer and the output of the VG-RAM WNN neural layer respectively. Firstly, each neuron extracts a binary input vector I from the input layer, via its set of synapses S (one bit per synapse). Secondly, the expected output label t is set in the output of the corresponding neuron in the neural layer. Finally, this input-output pair $L = (I, t)$ is subsequently stored into the neuron's *look-up table* (see Table 3.1).

During test, an input pattern is set in the input layer and each neuron extracts a binary input vector I from the given input pattern via its set of synapses S. The neurons subsequently use I to search and find, in their look-up tables, the input I_j, belonging to the learned input-output pairs $L_j = (I_j, t_j)$ that is the closest to the I vector extracted from the input layer. Finally, the output of the neuron receives the label value t_j of this L_j input-output pair. In case of more than one pair L_j with an input I_j at the same minimum distance of the extracted input I, the output value t_j is randomly chosen among them.

Table 3.1: VG-RAM WNN neuron lookup table.

Lookup Table	s_1	s_2	s_3	Y
L_1	1	1	0	t_1
L_2	0	0	1	t_2
L_3	0	1	0	t_3
	↑	↑	↑	↓
input	1	0	1	$\mathbf{t_2}$

Table 3.1 shows the lookup table of a VG-RAM WNN neuron with three synapses (s_1, s_2 and s_3). This lookup table contains input-output pairs $L_j = (I_j, t_j)$, which were stored during the training phase (L_1, L_2 and L_3). During the test stage, when an input vector (input) is presented to the network, the VG-RAM WNN test algorithm calculates the distance between this input vector and each

input of the input-output pairs stored in the lookup table. In the example of Table 3.1 the Hamming distance from the input to entry L_1 is two, because both s_2 and s_3 bits do not match the input vector. The distance to entry L_2 is one, because s_1 is the only non-matching bit. The distance to entry L_3 is three, as the reader may easily verify. Therefore, for this input vector, the algorithm evaluates the neuron's output, Y, as class 2, since it is the output value stored in entry L_2.

It is important to note that the Hamming distance between two binary patterns can be efficiently computed at machine code level in current 64-bit CPUs and GPUs of personal computers using two instructions: one to identify the bits that differ in 64-bit segments of the two binary patterns, i.e. a bit-wise exclusive-or instruction; and another to count these bits, i.e. a population count instruction.

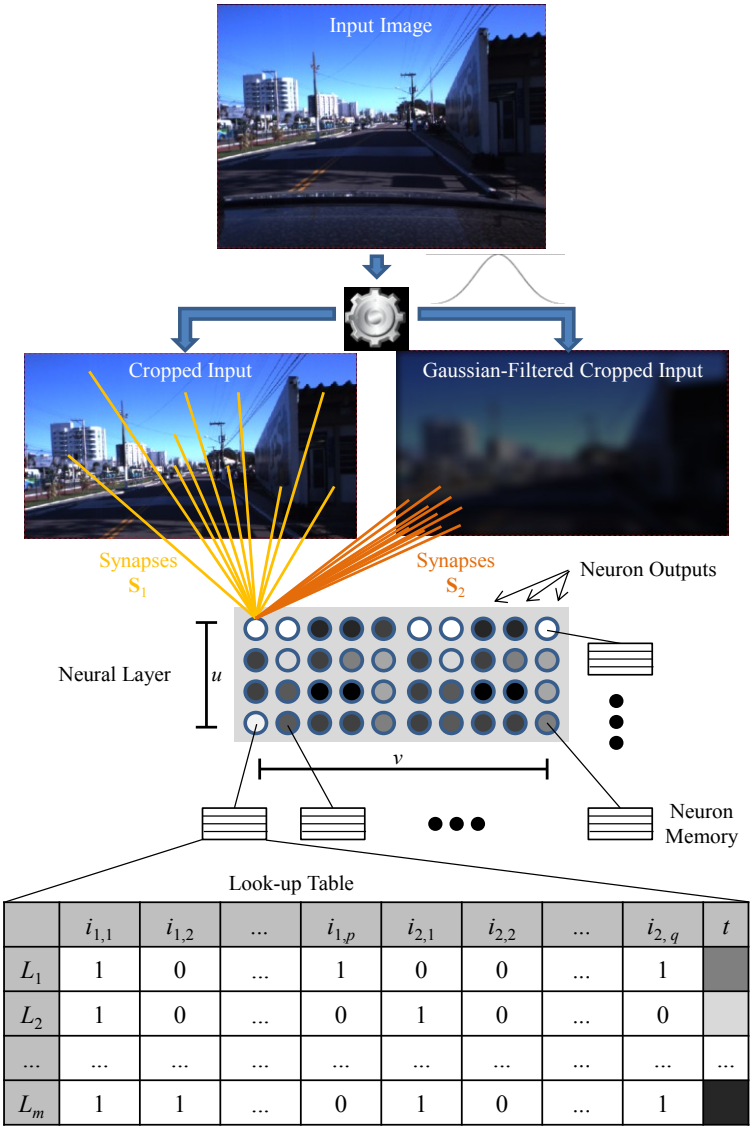
3.2 VG-RAM Image-Based Mapping (VIBM)

3.2.1 VIBM Architecture

The VIBM subsystem employs a VG-RAM WNN architecture that captures holistic and feature-based aspects of input images by using two different synaptic interconnection patterns. Figure 3.2 shows an overview of the VIBM subsystem. VIBM uses a single Neural Layer with $u \times v$ VG-RAM WNN neurons with m-size memory. This Neural Layer is connected to two input layers, *(i)* Cropped Input and *(ii)* Gaussian-Filtered Cropped Input, according to two different synaptic interconnection patterns, $\mathbf{S}_1$ and $\mathbf{S}_2$, respectively. $\mathbf{S}_1 = \{s_{1,1}, \ldots, s_{1,p}\}$ and $\mathbf{S}_2 = \{s_{2,1}, \ldots, s_{2,q}\}$ are subsets of $\mathbf{S} = \{s_{1,1}, \ldots, s_{1,p}, s_{2,1}, \ldots, s_{2,q}\}$, i.e., $\mathbf{S} = \mathbf{S}_1 \cup \mathbf{S}_2$, where $\mathbf{S}$ is the set of synapses of each neuron of the VIBM's Neural Layer.

Each neuron samples the Cropped Input and the Gaussian-Filtered Cropped Input in two different ways: holistically, with $\mathbf{S}_1$; and feature-based, with $\mathbf{S}_2$. The set of synapses $\mathbf{S}_1$ samples the Cropped Input holistically because it is defined according to a uniform random interconnection pattern that covers

the whole Cropped Input; while $\mathbf{S}_2$ samples the Gaussian-Filtered Cropped Input featured-based because it is defined according to a Normal distribution centered in the position of the neuron mapped to this input (see Figure 3.2 and [SOU08] for details about the feature-based synaptic interconnection pattern).



	$i_{1,1}$	$i_{1,2}$	...	$i_{1,p}$	$i_{2,1}$	$i_{2,2}$	...	$i_{2,q}$	t
L_1	1	0	...	1	0	0	...	1	
L_2	1	0	...	0	1	0	...	0	
...	...	...	...	...	...	...	...	...	...
L_m	1	1	...	0	1	0	...	1	

Figure 3.2: Illustration of the VIBM subsystem. VIBM employs a $u \times v$ VG-RAM WNN Neural Layer of neurons with m-size memory. Each neuron is connected to two processed versions of the Input Image (Cropped Input and Gaussian-Filtered Cropped Input) through two sets of synapses, S_1 and S_2 (exemplified for one neuron in yellow and orange respectively). $S_1 = \{s_{1,1}, \ldots, s_{1,p}\}$ and $S_2 = \{s_{2,1}, \ldots, s_{2,q}\}$ are subsets of $S = \{s_{1,1}, \ldots, s_{1,p}, s_{2,1}, \ldots, s_{2,q}\}$, i.e., $S = S_1 \cup S_2$, where S is the set of synapses of each neuron. This set of synapses samples the neuron's inputs as a vector of bits $I = \{i_{1,1}, \ldots, i_{1,p}, i_{2,1}, \ldots, i_{2,q}\}$. The Neural Layer shows an example of activation pattern based on the binary input vectors I and labels t of the learned pairs $L = (I, t)$. Each neuron responds with the label t_j associated with the input I_j that is the closest to the binary input vector I extracted from the Cropped Input and the Gaussian-Filtered Cropped Input. The labels t are indexes to geo-tagged images.

The synaptic mapping function that maps non-binary image pixels to

binary values is a *minchinton cell type* [MIT98] that works as follows. Each pixel is treated as an integer $y = b \times 256 \times 256 + g \times 256 + r$, where b, g, and r are the blue, green and red color channels. The non-binary pixel value y read by each synapse is subtracted from the non-binary pixel value y read by the subsequent synapse in the set of synapses of each neuron, $\mathbf{S} = \{s_{1,1}, \ldots, s_{1,p}, s_{2,1}, \ldots, s_{2,q}\}$. The value read by the last synapse, $s_{2,q}$, is subtracted from the value read by the first, $s_{1,1}$. If a negative value is obtained, the bit corresponding to that synapse is set to one; otherwise, it is set to zero.

The two input layers, Cropped Input and Gaussian-Filtered Cropped Input, are processed versions of the Input Image. While the Cropped Input is simply a region of interest defined in the input image, the Gaussian-Filtered Cropped Input is the result of a Gaussian filter applied to this region of interest (see Figure 3.2 for an example).

The region of interest was defined in order to remove irrelevant pixel information from the input image. In our case, the bottom of the image is cropped out to eliminate static part of the car roof visible in the field of view of a mounted-on camera. The Gaussian filter, in the other hand, is used as a low-pass image filter. Since a feature-based synaptic interconnection pattern is used to sample this input layer, high-frequency attenuation is necessary to remove spurious high-frequency information irrelevant for localization.

3.2.2 Mapping

The VIBM subsystem learns images from the environment and associated global poses and 3D landmarks (i.e., the Neural Map). Let global_pose_j be the global pose of the image_j and U_j be the set of 3D landmarks of image_j. Let also $\mathbf{T} = \{T_1, \ldots, T_j, \ldots, T_{|\mathbf{T}|}\}$ be a set of triplets $T_j = (\text{image}_j, \text{global\_pose}_j, \text{U}_j)$ presented to VIBM. In the mapping phase (or training), the image_j of each triplet T_j is set as the VIBM's Input Image and the corresponding index j is copied to the output of each neuron of VIBM's Neuron Layer. Then, all neurons are trained to output j when sampling from image_j via Cropped Input and Gaussian-Filtered Cropped Input images.

The contents of all neurons' memories - the Neural Map – are dumped to a file, for a posteriori usage. In the localization phase (or test) (Section 3.3 and Section 3.4), the map file is loaded to the neurons' memories and the index j learned by the neurons can be used for recovering $pose_j$, $image_j$ or U_j.

In the following section, we describe how the characteristic points required by the VIBM's learning procedure are detected. In Section 4.1 we describe how the global poses are computed.

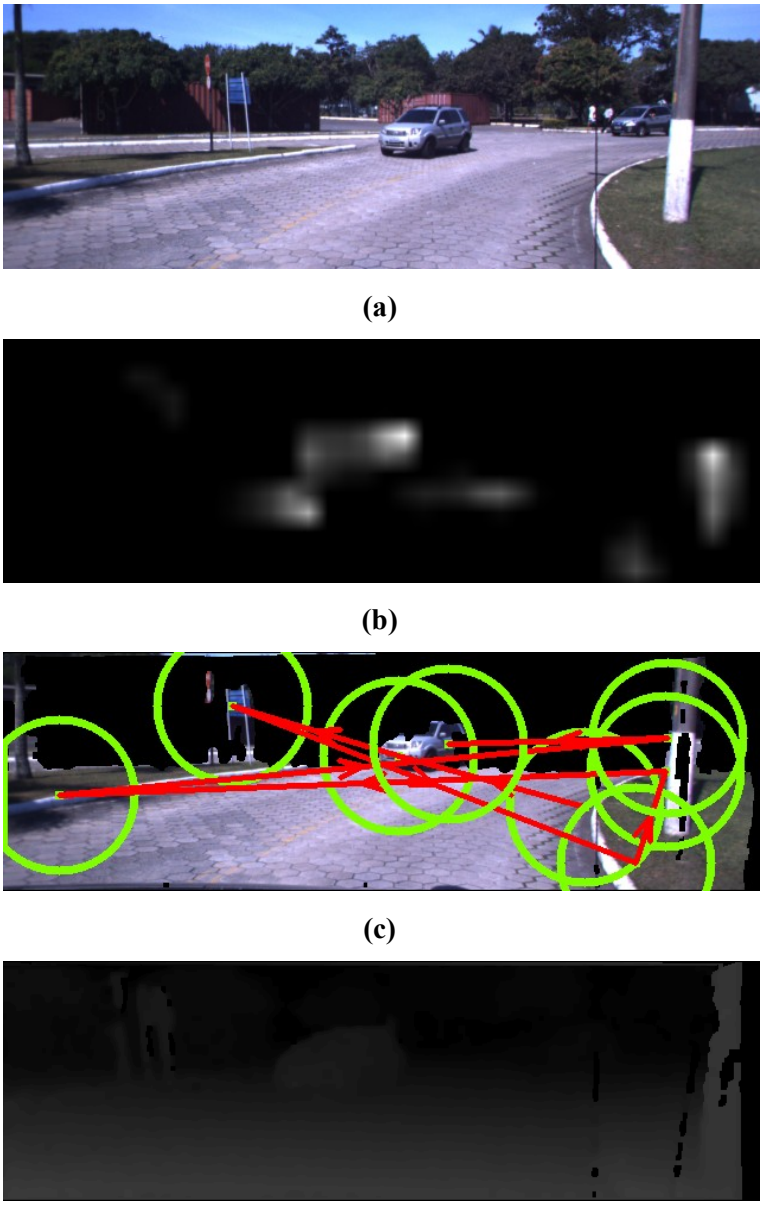
3.2.3 Detection of Characteristic Points

To detect image characteristic points, the VIBM subsystem employs the iLab Neuromorphic Tookit Vision C++ Tool (iNVT, pronounced "invent") [ITT98, NAV05]. iNVT is a set of C++ classes for the development of neuromorphic models of vision. Particularly, we use the iNVT neuromorphic model that is inspired in visual attention. This model estimates which elements of a scene are likely to attract the attention of human observers. These elements are considered the characteristic points or saliencies of an image.

Figure 3.3 illustrates the detection of characteristic points on an image. Given an input image (Figure 3.3 (a)), the iNVT's visual attention model computes an initial saliency map (Figure 3.3 (b)). This saliency map is a combination of feature-maps which represent local discontinuities of an image, in the modalities of color, intensity and orientation. A winner-takes-all neural network detects the points of highest contrast in the salience map and draws the focus of attention towards these locations, which are considered saliencies (Figure 3.3 (c)). For each shift of the focus of attention, an inhibition process is performed in order to prevent that saliency to be detected twice, basically this inhibition process works by erasing the found saliency in the saliency map. After this inhibition process occurs, the saliency map is update and the above steps (detection of highest contrast, shift of attention and inhibition) are repeated until a certain number of saliencies is computed.

To compute the three dimensional positions of detected saliencies, VIBM employs the Library for Efficient Large-scale Stereo Matching (LIBELAS)

[GEI10]. Given a pair of stereo images, LIBELAS computes a depth map (Figure 3.3 (d)).



(a)

(b)

(c)

(d)

Figure 3.3: (a) Scene image. (b) Initial saliency map computed by iNVT. (c) Image saliencies detected by iNVT. (d) Depth map computed by LIBELAS.

A depth map is an image where each pixel represents the distance between the camera position and the surface of objects from a world scene. Using the information of distance stored in the depth map, and the stereo camera's projective parameters, VIBM can compute the three dimensional positions of saliencies (3D landmarks).

3.3 VG-RAM Image-Based Global Localization (VIBGL)

3.3.1 Global Localization

To perform global localization, the VIBGL subsystem uses the same VIBM's architecture. As a matter of fact, the VIBGL is only the representation of the VIBM's test phase.

At initialization, the VIBGL subsystem firstly loads the map of the environment – the Neural Map – stored in the map file to its neurons' memories (Section 3.2).

At runtime, given a query image, VIBGL infers a global pose based on the previously acquired knowledge. The query image is set as VIBGL's Input Image and all neurons compute their outputs, which are indexes (32-bit integers). Each neuron infers an index based on the input binary vectors extracted by their synapses. The number of votes for each index is counted and the network outputs the index *j* with the largest count. The index *j* is used to recover the $image_j$, $global\_pose_j$ or the 3D landmark set U_j, that are outputted by VIGBL.

3.4 VG-RAM Image-Based Position Tracking (VIBPT)

In order to perform activities of interest, autonomous robots need to know its initial pose (global localization) and to keep track of its new poses over time with small uncertainty (position tracking). The VIBGL, subsystem of VIBML, efficiently solves the global localization problem, but it does not solve the position tracking problem, because the uncertainty about the global pose is not negligible.

The major restriction of VIBGL is that it estimates the robot's global pose using previously acquired knowledge – the map – without performing any correction on the estimated global pose. When VIBGL is building its internal representation of the environment (using the VIBM architecure), it learns that a particular $image_j$ was captured at $global\_pose_j$. After that, in localization phase, when another arbitrary $image_i$, (similar to the $image_j$) is

presented to VIBGL, it outputs that the inferred image pose is exactly global_pose$_j$. Nevertheless, this is not necessarily true, since the image$_i$ may have been captured at global_pose$_i$, that is slightly different of the VIBGL's outputted global_pose$_j$ (see Figure 3.4). In this way, the VIBGL's estimated global_pose$_j$ may contain a displacement error that needs to be corrected to best approximate the real global_pose$_i$.

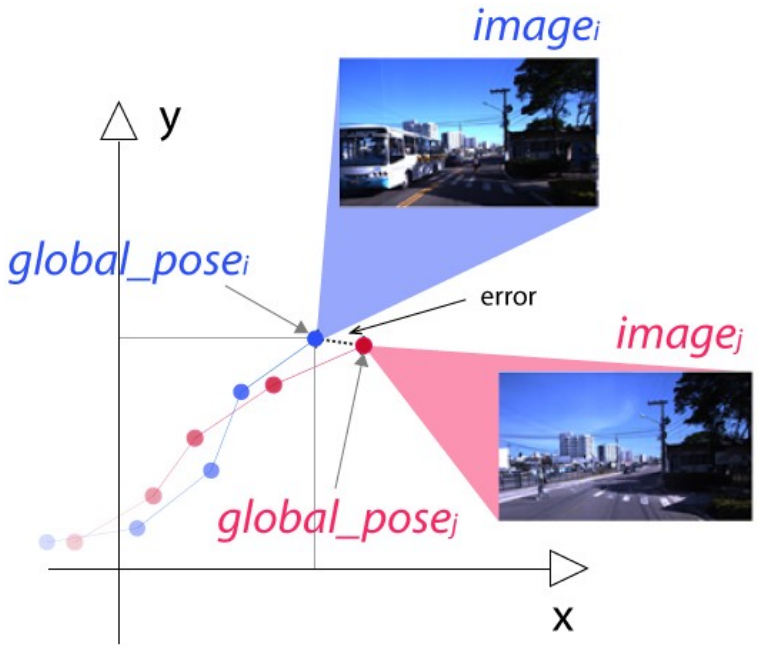


Figure 3.4: Error in the global pose of an image estimated by VIBGL. Given a query image$_i$, VIBGL outputted the global_pose$_j$ associated with image$_j$. Nevertheless, image$_i$ might be captured at a global_pose$_i$ slightly different from global_pose$_j$.

To overcome this problem and turn VIBML into a full localization system, we built the VIBPT subsystem, which integrates the VIBGL's estimated global poses with the matching of landmarks previously stored in the map in order to provide a more reliable and precise robot's pose. For that, VIBPT employs an Extendend Kalman Filter (EKF) that operates in two steps: the state prediction step and the state correction step. In the state prediction step, EKF uses our robot's motion model and velocity information to estimate a local pose. In the state correction step, EKF firstly corrects the local pose by the matching of landmarks previously stored in the map, and subsequently fuses the corrected local pose with the global pose estimated by VIBGL, which ensures a local pose error bounded by the global pose error.

3.4.1 Extended Kalman Filter (EKF)

The Kalman Filter (KF) is a recursive filter that estimates the state of a linear system [THR05]. At a given time, it uses its previous knowledge about the system's state and sensor measurements to compute a predicted value of the state and the covariance matrix of the estimation error. The Extended Kalman Filter (or EKF) is a KF that linearizes the non-linear dynamics about the system around the previous state estimates [THR05]. It is a sub-optimal method and is reliant on the noise of the system being Gaussian distributed.

Table 3.2 shows the EKF algorithm. The EKF represents the system's state *X* at time *t* by means of the mean μ_t and the covariance Σ_t of a multivariate Gaussian distribution. In general, on each iteration, the EKF tries to keep the system's state estimate updated, by computing consecutive state predictions and corrections steps.

Table 3.2: The EKF algorithm [THR05].

1: **Algorithm Extended_Kalman_Filter** $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$
2: $\bar{\mu}_t = g(u_t, \mu_{t-1})$
3: $\bar{\Sigma}_t = G_t \, \Sigma_{t-1} \, G_t^T + R_t$
4: $K_t = \bar{\Sigma}_t \, H_t^T \, (H_t \, \Sigma_t \, H_t^T + Q_t)^{-1}$
5: $\mu_t = \bar{\mu}_t + K_t \, (z_t - h(\bar{\mu}_t))$
6: $\Sigma_t = (I - K_t \, H_t)\bar{\Sigma}_t$
7: return μ_t, Σ_t

In the state prediction step, the EKF predicts the state estimate by employing a continuous nonlinear function, $g(u_t, \mu_{t-1})$, that governs the system state transition model (lines 1 and 2 of Table 3.2). This model describes how the system state $X = \binom{\mu}{\Sigma}$, evolves over time. Given a command u_t and the previous state mean μ_{t-1}, the function $g(u_t, \mu_{t-1})$ computes the predicted state mean $\bar{\mu}_t$. In the correction step (lines 4 to 6 of Table 3.2), the EKF receives observations z_t as input and uses them to correct the predicted state mean $\bar{\mu}_t$ by comparing the observations z_t with the expected measurements computed by the system's measurement model, that is

governed by the non-linear function $h(\bar{\mu}_t)$.

The system's covariance Σ_{t-1} is also updated in the process. At prediction, EKF uses the Jacobian G_t plus an additive Gaussian noise with zero mean, R_t, to update the previous covariance Σ_{t-1} to a new covariance, $\bar{\Sigma}_t$. The Jacobian G_t is the derivative of the function g with respect to the previous state $X_{t-1} = \begin{pmatrix} \mu_{t-1} \\ \Sigma_{t-1} \end{pmatrix}$, and evaluated at the command u_t and the previous state mean μ_{t-1} (line 2). At correction, EKF uses the Jacobian H_t plus an additive Gaussian noise with zero mean, Q_t, to correct the predicted $\bar{\Sigma}_t$. The Jacobian H_t is the derivative of the function h (measurement model) with respect to the robot location, and evaluated at the predicted mean $\bar{\mu}_t$ (line 6). Q_t is an additive Gaussian noise with zero mean that represents the sensor's noise.

3.4.2 Localization with EKF

In this work, we used the EKF in the context of mobile robot localization [THR05] and implemented it employing the Bayesian Filtering Library (BFL [KLA01]).

The system state transition model was defined by means of the velocity motion model of an autonomous car. This velocity motion model considers the kinematics of a car-like robot and assumes that we can control it through translational velocity and steering wheel angle commands.

The system measurement model was split in two components: a linear measurement model and a landmark measurement model [THR05, SIM06]. Firstly, we used a simple linear measurement model with additive Gaussian noise to fuse the global pose (estimated by VIBGL) with the local pose (estimated by VIBPT in the EKF state prediction step). In this way, VIBPT can guarantee that the local pose does not drift so much over time and the uncertainty about the local pose is bounded by the global pose error. Subsequently, we used the landmark measurement model to update the previous global correction, by matching the detected landmarks observed in the sensor data (3D observations) along a trajectory, with landmarks stored in

the map (3D landmarks).

3.4.2.1 State Prediction Step

Our EKF implementation employs the velocity motion model of a car-like robot in the state prediction step.

Let x and y be the car's location, given by the midway of the two rear wheels; θ the car's orientation; L the distance between the front and rear wheels' axles; v the car's translational velocity; φ the steering wheel angle, given by the average of the angle of the right and left front wheels; as illustrated in Figure 3.5.

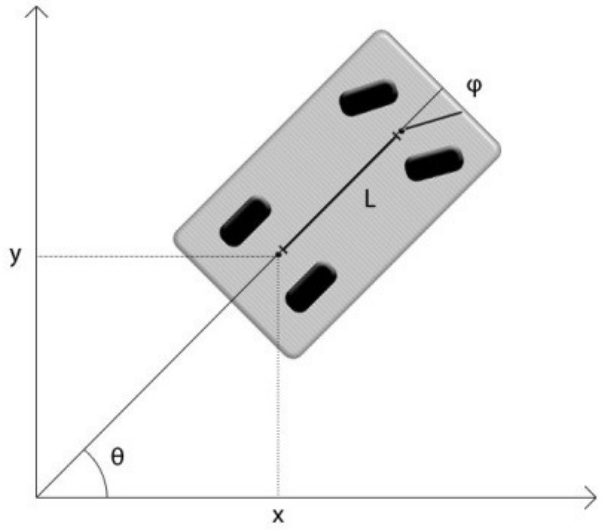


Figure 3.5: Parameters of the velocity motion model of a car-like robot.

Also, let $X_t = (x, y, \theta)$ be the state of the car at time t and $u_t = (v, \varphi)$ the control command at time t. So, after the small Δt time interval, the car will be at state $X_{t+1} = (x', y', \theta')$ given by the g function:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \Delta t\ v \cos\theta \\ \Delta t\ v \sin\theta \\ \Delta t\ v\ \frac{\tan\varphi}{L} \end{pmatrix} + \mathcal{N}(0, R_t), \qquad (1)$$

where $\mathcal{N}(0, R_t)$ is a Gaussian distribution with zero mean and covariance R_t, which represents the noise of the velocity motion model.

To compute $u_t = (v, \varphi)$, the VIBPT subsystem uses velocity and

steering angle information computed from images using visual odometry (Section 3.4.3).

3.4.2.2 State Correction Step

To perform the correction step, we firstly used a simple linear measurement model with additive Gaussian noise to fuse the global (VIBGL) and local (VIBPT) predicted poses [SIM06].

Secondly, we employed the landmark measurement model [THR05] to compare the range-and-bearing of map-stored landmarks (3D landmarks) with landmarks observed in the sensor data (3D observations). To compute the Euclidean distance (range) and the orientation angle (bearing) between the robot's local pose and the expected 3D landmark's position in the map we used the Equation (2), that represents the function h

$$\underbrace{\begin{pmatrix} r_t^i \\ \phi_t^i \end{pmatrix}}_{z_t^i} = \begin{pmatrix} \sqrt{(m_x^i - x)^2 + \left(m_y^i - y\right)^2} \\ \text{atan2}\left(m_y^i - y,\, m_x^i - x\right) - \theta \end{pmatrix} + \mathcal{N}(0, Q_t), \qquad (2)$$

where $(m_x, m_y)^T$ is the coordinates of the i-th landmark in the map, detected at time t; (x, y) is the coordinates of the robot's local position, θ is the robot's orientation; and $\mathcal{N}(0, Q_t)$ is a Gaussian distribution with zero mean and covariance Q_t, which represents the sensor's noise.

Finally, we computed the mean correction by updating it proportionally to the displacement between the i-th 3D landmark's measurement z_t^i computed using (2) and the 3D observations currently made by sensors (line 5 of Table 3.2).

In order to compute the correspondences between landmarks detected by sensors (3D observations) with landmarks stored in the map (3D landmarks), the VIBPT subsystem used a visual search approach based on VG-RAM WNN [SOU13] (Section 3.4.4).

3.4.3 Visual Odometry

The VIBPT subsystem employs the Library for Visual Odometry 2 (LIBVISO2) [GEI11] in order to compute $u_t = (v, \varphi)$. LIBVISO2 estimates the relative displacement between two consecutive positions of a camera over time using the stereo images captured in these positions. Given the relative displacement between two consecutive camera poses, $u_t = (v, \varphi)$ can be computed by:

$$v = \frac{\sqrt{\delta_x^2 + \delta_y^2}}{\Delta t} \text{ and} \quad (3)$$

$$\varphi = \text{atan2}\left(L\frac{\delta_\theta}{\Delta t}, |v|\right), \quad (4)$$

where δ_x and δ_y are the relative displacements in the x and y coordinates, respectively, and δ_θ is the displacement in the orientation.

3.4.4 Visual Search of Landmarks

Figure 3.6 shows how the VIBPT system performs the matching between the 3D landmarks previously stored in the Neural Map with the 3D observations currently made by the robot.

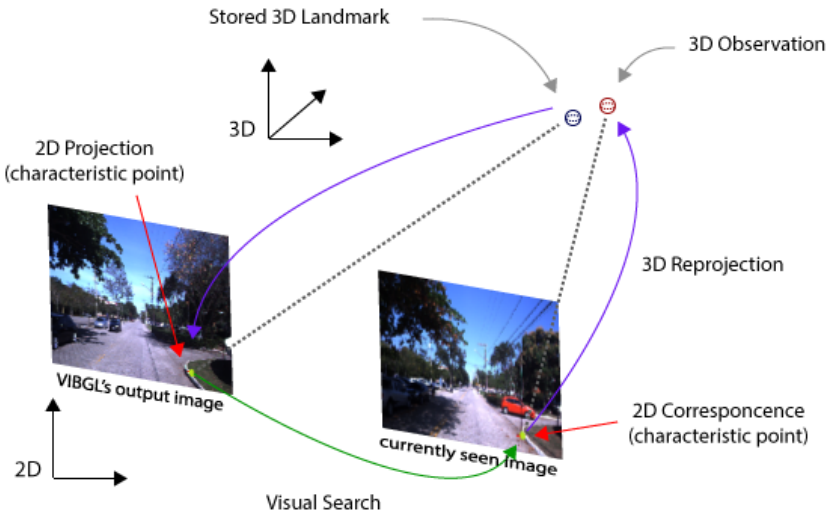


Figure 3.6: Visual Search of map-stored 3D landmarks in the image currently observed by the robot.

Firstly, the VIBPT subsystem consults VIBGL for the most similar image (left image in Figure 3.6) and respective 3D landmarks (blue sphere in

Figure 3.6), to the image currently seen by the robot (right image in Figure 3.6).

Then, VIBPT reprojects the 3D landmarks outputted by VIBGL back to the camera's coordinate system (left-blue arrow in Figure 3.6) (2D coordinates of characteristic points) and searches for these characteristic points in the image seen by the robot, using a visual search approach based on VG-RAM WNN [SOU13] (green arrow in Figure 3.6).

Once the correspondences for each characteristic point is found, VIBPT computes their three dimensional positions, the 3D observations (red sphere in Figure 3.6), using the distance information from a depth map computed using the LIBELAS stereo matching algorithm [GEI10] (right-blue arrow in Figure 3.6).

Using the map-stored 3D landmark and its correspondence found by VG-RAM Visual Search, VIBPT computes two measurement vectors: the expected measurement vector (3D landmarks), represented by the distance and angle between the robot's local pose and the pose of the landmark stored in the map. And the observation measurement vector (3D observations), represented by the distance and angle between the robot's pose and the 3D landmark found correspondence. Finally, the expected measurement and observation measurement vectors are used by the landmark measurement model via the EKF's measurement model to correct the robot poses proportionally to the displacement between the two above mentioned vectors.

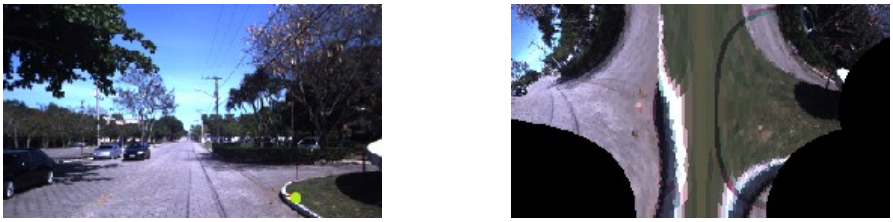
3.4.4.1 Context Application

Figure 3.7 shows an example of a training instance of our VG-RAM WNN architecture for visual search.

In Figure 3.7, the network is trained to detect the curb of the street on the image. Figure 3.7(a) shows the training image with the centre of attention marked with a green dot; Figure 3.7(b) shows the log-polar mapping of the VG-RAM WNN's input onto the network neural layer; and Figure 3.7(c) shows the output of the neural layer after training.

As the Figure 3.7(c) shows, neurons with receptive field over or near the

center of attention are trained to produce outputs with values higher than zero (white or gray), while those with receptive field far from the center of attention are trained to output zero (black).

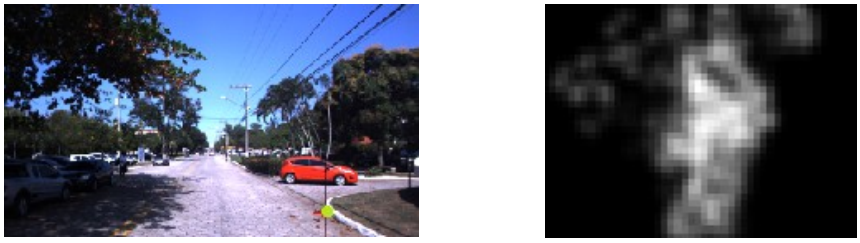


(a) **(b)**

(c)

Figure 3.7: Example of a training instance of the VG-RAM WNN architecture for visual search. (a) Training image and characteristic point to search for (green dot). The Log-Polar for the Training Image. (c) Neurons activation.

Figure 3.8 shows an example of a test instance of our VG-RAM WNN architecture for visual search, where neurons of the network, trained to detect the curb, generate their outputs according to the image region monitored by their receptive fields. Figure 3.8(a) shows the test image with the found centre of attention marked with a green dot; Figure 3.8(b) shows the output of the VG-RAM WNN's neural layer. Figure 3.8(b) shows that neurons with the centre of their receptive fields over or near the centre of attention generate higher outputs.



(a) (b)

Figure 3.8: Example of a test instance of our VG-RAM WNN architecture for visual search.

3.4.5 Outliers Removal

Although the VIGBL subsystem usually estimates global poses with an acceptable accuracy, it might sometimes predict a global pose that is far from the actual robot´s global pose. Such a wrong prediction causes a bad measurement integration in the VIBPT's linear measurement model. To minimize this issue, we choose the best global pose estimation, bg, to be used in the linear measurement model among all global poses, g, estimated by VIBGL. The choice is based on how close the global pose is from the previous local pose, p, estimated by VIBPT as in Equation (5):

$$bg = \underset{g}{argmin}\left(\sqrt{(g_x - p_x)^2 + \left(g_y - p_y\right)^2}\right) \qquad (5)$$

Hence, the smaller the Euclidean distance between the VIBGL's estimated global pose, g, and the previous VIBPT's estimated local pose, p, is, the greater are the chances of g being the best global pose estimation, bg. If the distance between these two poses is larger than a pre-defined threshold, there is a high chance of the estimated global pose, g, being an outlier and, therefore, it is discarded by the system. In this implementation, VIBGL returns three guesses for the global pose g (i.e., the three most voted poses) to choose the best estimation for the VIBPT correction step, as described above.

Chapter 4

Experimental Methodology

In this chapter, we present the experimental methodology used to evaluate the VIBML system. We start by presenting the autonomous vehicle platform used to acquire the datasets, follow by describing the CARMEN Robot Navigation Toolkit employed to implement the VIBML system and the datasets used for the experiments. We finish by describing the methodology and metrics used in the experiments.

4.1 Autonomous Vehicle Platform

We collected the data to evaluate the VIBML system's performance using the Intelligent and Autonomous Robotic Automobile – IARA (Figure 4.1). IARA is an experimental robotic platform based on a Ford Escape Hybrid that is currently being developed at *Laboratório de Computação de Alto Desempenho* – LCAD (High-Performance Computing Laboratory – www.lcad.inf.ufes.br) of *Universidade Federal do Espírito Santo* – UFES (Federal University of Espírito Santo – Brazil).

Our robotic platform has several high-end sensors, including: two Point Grey Bumblebee XB3 stereo cameras and two Bumblebee 2 stereo cameras; one Light Detection and Ranging (LIDAR) Velodyne HDL 32-E; and one

GPS-aided Attitude and Heading Reference System (AHRS/GPS) Xsens MTiG (Figure 4.1). To process the data coming from the sensors, the platform has four Dell Precision R5500 (2 Intel Xeon 2.13 GHZ, 12 GB RAM, 2 SSDs of 120GB on RAID0 and GPU cards Tesla C2050). We implemented many software modules for IARA that currently allows for its autonomous operation (such as modules for mapping, localization, obstacle avoidance, navigation, etc.; see video of IARA autonomous operation at http://youtu.be/zE7np6tgCHc and videos about other IARA's software modules at http://www.youtube.com/user/lcadufes).

Figure 4.1. Intelligent and Autonomous Robotic Automobile (IARA) with the mounted-on Point Grey Bumblebee XB3 camera (marked in green) used in experiments. Learn more about IARA at www.lcad.inf.ufes.br.

To build the datasets used in this work, we used IARA's frontal Bumblebee XB3 left camera to capture images (640x480 pixels), and IARA's Occupancy Grid Mapping - Monte Carlo Localization (OGM-MCL) system to capture associated global poses.

The OGM-MCL system works by fusing visual odometry, Global Positioning System (GPS) pose and Inertial Measurement Unit (IMU) data from IARA's sensors into a precise fused odometry using a Particle Filter, and then localizes the robot on a previously created occupancy grid map. The OGM-MCL system uses the fused odometry and the vehicle's motion model (suitable for vehicles with Ackermann steering) to predict the vehicle's pose and correct it by performing the matching between the IARA's Velodyne

HDL-32 data with a previously created occupancy grid map. The global poses computed by IARA's OGM-MCL system have precision of about 0.5m.

4.2 CARMEN Robot Navigation Toolkit

All the VIBML system's modules were implemented using the CARMEN Robot Navigation Toolkit. CARMEN is an acronym for the popular and widely used "Carnegie Mellon Robot Navigation Toolkit", a collection of open source software (http://carmen.sourceforge.net/home.html), designed at the Carnegie Mellon University (CMU) to control mobile robots.

The toolkit guaranteed to its developers the victory in the Defense Advanced Research Projects Agency (DARPA) Grand Challenge 2005 (http://archive.darpa.mil/grandchallenge05) and the second place in DARPA Urban Challenge (http://archive.darpa.mil/grandchallenge/index.asp). CARMEN allows abstracting most of the implementation details of a robotic system that incorporates sensors, algorithms for planning, navigation and control, freeing the programmer to focus on issues of the highest level.

CARMEN is modular, service oriented and provides basic primitives for robot navigation, including: base and sensor control, registration, detection and obstacle avoidance, localization, path planning and mapping.

CARMEN enables the development of systems consisting of multiples executable programs (or modules) that communicate together according to the publish-and-subscribe paradigm. As stated by this paradigm, a sensor module, for instance, can be implemented by a separate executable program that sends (publish) messages with sensor's data for any modules that sign (subscribe to) these messages. A filter module can sign messages of various modules, manipulate them with algorithms of interest, and post messages with their results for various other modules that request them.

A module that publishes a message does not need to know who receives it; thus avoids problems like dead lock and starvation that hinder the programming of distributed systems (autonomous robot control systems are

inherently distributed).

4.3 Datasets

For the experiments, we have used two laps data acquired in different dates. Basically, for each lap, IARA was driven with an average speed of about 30 km/h around the UFES campus. A full lap around the university campus has an extension of about 3.57 km (Figure 4.2). During the laps, image and global pose data were synchronously acquired.

Figure 4.2: Full lap around the university campus with an extension of about 3.57 kilometers. Source: Google Maps (http://maps.google.com.br).

The first lap data was recorded in October 3rd 2012 (UFES-2012), while the second lap data was recorded in April 18th 2014 (UFES-2014). The difference in days between the recording of the first and the second lap data is almost two years. Such time difference resulted in a challenging testing scenario since it captured substantial changes in the campus environment. Such changes includes differences in traffic conditions, number of pedestrians, and changes in lighting condition. Also, there were substantial building infrastructure modifications alongside the roads in between dataset recording.

To evaluate the effect of learning different numbers of images (the more images the VIBGL, subsystem of VIBML, learns, the more labels it has to

differentiate) the UFES-2012 lap data were sub-sampled at four different intervals: 1 meter, 5 meters, 10 meters, and 15 meters. After sub-sampling the UFES-2012, four datasets were created: 1-meter spacing dataset with a total of 2,223 image-pose pairs, a 5-meters dataset with a total of 444 image-pose pairs, a 10-meters dataset with 222 image-pose pairs, and a 15-meters dataset with 148 image-pose pairs. The UFES-2014 dataset was not sub-sampled, since it was only used for test purposes. All datasets mentioned above are available at: http://www.lcad.inf.ufes.br/log.

4.4 Metrics

In order to validate our system, we have run a set of localization experiments. In all experiments, the training and test datasets were from different dates, except in the Localization Noise experiment of the VIBPT subsystem, where we used the same dataset for training and test.

In other experiments, the sub-sampled datasets from UFES-2012 were used to teach the VIBM subsystem about a trajectory (training the system), and the UFES-2014 dataset was used to test the performance of the system by comparing VIBML's estimated poses with the output poses from IARA's OGM-MCL system, along the learned trajectory.

4.4.1 Global Localization Metrics

In order to evaluate the VIBGL subsystem, we used two distinct metrics to measure the VIBGL's classification accuracy and the VIBGL's positioning error.

Firstly, we measured the VIBGL's classification accuracy by means of how many image-pose pairs the VIBGL subsystem estimates correctly. Secondly, we measured the VIBGL's positioning error by means of how close de VIBGL's estimated poses p_i are to the poses p_j estimated by the IARA's OGM-MCL system. For this, we compute the Mean Absolute Error (MAE) of the Euclidean distance between these two set of poses. The MAE is given by

Equation (6)

$$MAE = \frac{1}{n} \sum_{i=1,\ j=1}^{n} |p_i - p_j| \quad (6)$$

where, n is the number of image-pose pair compared, p_i is the VIBGL's estimated pose and p_j is the pose estimated by the IARA's OGM-MCL system.

4.4.2 Position Tracking Metrics

In order to evaluate the VIBPT sub-system, we firstly, measured the VIBPT's positioning error by means of how close de VIBPT's estimated poses p_i are to the poses p_j estimated by the IARA's OGM-MCL system. For this, we employed the MAE metric used in global localization (Equation (6)) to compute the average distance between these two set of poses. In addition, we compared the VIBML performance improvement when using positioning tracking rather than global localization only.

Secondly, we compared the VIBPT and the OGM-MCL systems by measuring the localization noise and the localization displacement regarding the IARA's OGM-MCL pose estimates in a full trajectory around the UFES' campus and compare it against to the localization noise and the localization displacement regarding the VIBPT's pose estimates.

To measure the localization noise of each one of the localization systems, we basically run a set of experiments, using the same dataset (UFES-2012) for training and test. Firstly, we measured the Euclidean distance of the estimated poses between the experiments and calculate their standard deviations. Subsequently, we calculate the mean of these standard deviations using the Square Root of the Pooled (or weighted) Variances (SRPV [HEA10]), defined in Equation (7).

$$SRPV = \sqrt{\frac{1}{k} \sum_{i=1}^{k} \sigma_i^2} \quad (7)$$

where k is equal to the number of experiments performed and σ_i is the standard deviation of the Euclidean distance of the estimated poses between the experiment *i* and *i-1*.

To measure the localization displacement of each one of the localization systems, we run two experiments. The first one using the UFES-2012 dataset for training and test, and the second one using the UFES-2012 dataset for training and the UFES-2014 for test. Basically, in both experiments we recorded the trajectory estimated by the VIBPT and OGM-MCL system and we measured the MAE of the Euclidean distance (Equation (6)) between the estimated poses in the two experiments for both systems.

Chapter 5

Experiments

In this chapter, we show and discuss the outcomes of our experiments. We firstly present the experiments performed to evaluate VIBGL in three parts: classification accuracy, positioning error, and qualitative results. Subsequently, we present the experiments carried out to analyse VIBPT.

5.1 VIBGL

5.1.1 Classification Accuracy

This section shows the relationship between the amount of frames learned by the VIBGL system and its classification accuracy. We measured the system classification accuracy in terms of how close the VIBGL's estimated image-pose pair, $I_e = \{image_e, global\_pose_e\}$, is to the correct image-pose pair, $I_c = \{image_c, global\_pose_c\}$, for a given query image $I_q = \{image_q\}$. The image-pose pairs I_e and I_c belong to the training dataset, while the image I_q belongs to the test dataset. Ideally, I_e is equal to I_c if VibGL is correct in its estimate, since both image-pose pairs I_c and I_e belongs to the training dataset.

Figure 5.1 shows the classification accuracy results obtained using UFES-2012 dataset for training and UFES-2014 dataset for testing. The vertical axis represents the percentage of image-pose pairs I_e that were within

an established maximum number-of-frames distance from the image-pose pair I_c. The number-of-frames distance is equal to the amount of image-pose pairs that one has to go forward or backwards in the training dataset to find I_c from the corresponding I_e, and is represented as the horizontal axis. Finally, the curves of the graph of Figure 5.1 show how the accuracy increases with the allowed maximum number-of-frames distance for the different training datasets.

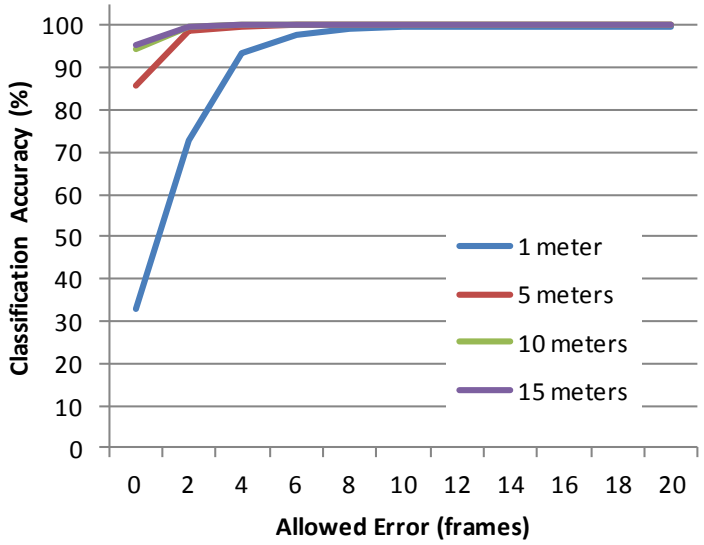


Figure 5.1. Classification accuracy for different maximum number-of-frames allowed using UFES-2012 dataset for training and UFES-2014 dataset for test.

As the graph of Figure 5.1 shows, VIBGL's classification accuracy increases with the maximum allowed number-of-frames and reaches a plateau at about 5 frames for all training datasets. However, for the UFES-2012 1-meter spacing training dataset, the VIBGL classification uncertainty is large in the beginning of the curve due to the similarity between images in the near-by image-pose pairs. If one does not accept any system error (number-of-frames allowed equal zero), the accuracy is only about 33% when the system is trained with the 1-meter spacing dataset. But, if one accepts as correct an image-pose pair up to 5 frames ahead or behind the correct image-pose pair, I_c, the accuracy increases to about 97%. On the other hand, when using a dataset with a larger spacing between image-pose pairs for training, the system accuracy increases more sharply. For example, when the system is trained

with the 5-meter spacing dataset, with an allowed number-of-frames equal to 1, the classification rate is about 85%.

Although the VIBGL might show better accuracy when trained with large-spaced datasets, the positioning error of the system increases. This happens because one frame of error for the 1-meter training dataset represents a much smaller error in meters than one frame of error with large-spaced training dataset (e.g., 10m).

5.1.2 Positioning Error

We performed experiments to evaluate the relationship between the spacing between image-pose pairs learned by VIBGL and the positioning error of its estimated poses compared to the IARA's OGM-MCL poses.

The results of these experiments are shown in Figure 5.2 as box-plots having mean, inter-quartile range and whiskers of the error distribution for the 1-meter, 5-meters, 10-meters and 15-meters training datasets. Box-plots are shown for the setup UFES-2012 as training and UFES-2014 for test.

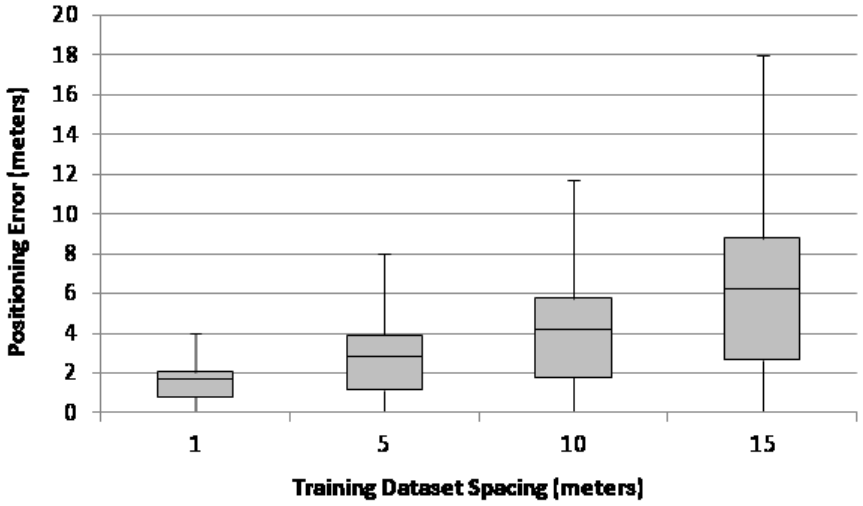


Figure 5.2. Positioning Error Distribution between I_e and I_q using the UFES-2012 dataset for training and the UFES-2014 dataset for testing.

The horizontal axis of Figure 5.2 shows training datasets spacing intervals, while the vertical axis shows the distance of the estimated image-pose pair I_e to the given image I_q. As the graph of Figure 5.2 shows, the positioning error increases as the spacing between training image-pose pairs

increases, but not linearly. The average error is larger than 1m for the 1-meter spacing training dataset, but smaller than the spacing of the other datasets. The performance of VIBGL with the 1-meter dataset can be explained by the fact that with more images, the VIBGL's VG-RAM WNN has more difficulty to differentiate between images.

5.1.3 Qualitative Results

To visualize the qualitative results for VIBGL's estimated positions, we extracted two samples of matched frames along the UFES campus: the first one having five true positive samples (Figure 5.3), and the second one having three false positive samples (Figure 5.4).

Figure 5.3. True positive qualitative results for VIBGL's frame estimation.

Figure 5.3 shows examples of true positive frames using the UFES-2012 dataset as training dataset. As it can be seen, the frames were matched despite changes in sunlight position and shadows (third and fifth), road infrastructure (first and fourth rows), car movements (second row) and loss of leaves on the trees (first, second and fifth rows).

Figure 5.4 shows examples of false positive frames using the UFES-2012 as training dataset. The system seems to fail at places with certain similarity, e.g., the sky-shape in the first and third rows looks the same. Moreover, in the third row, the two frames were captured in a place with similarities in the lane. Although this facts can explain those bad results, we must perform a deep investigation to understand why the VG-RAM WNN's neurons miss their estimates on those images.

An online demo video shows the VIBGL's performance on a complete lap around the university campus (see video at http://youtu.be/PMif-W6L2EY). In the video, we used the 1 meter-spacing UFES-2012 dataset for training and the 1 meter-spacing UFES-2014 dataset for testing.

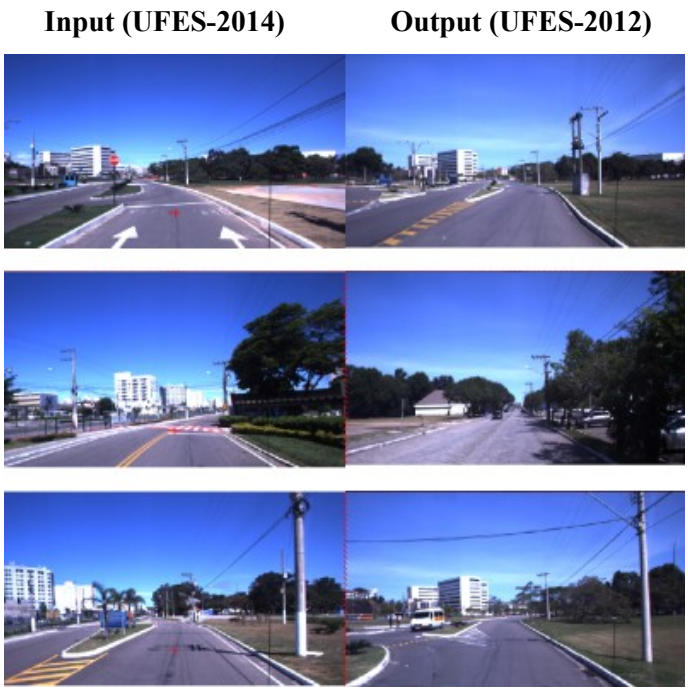


Figure 5.4. False positive qualitative results for VIBGL's frame estimation.

More information about the VIBGL subsystem and about VIBGL's

performance can be found in [LYR14].

5.2 VIBPT

5.2.1 Positioning Error

We measured the VIBPT's positioning error by means of how close de VIBPT's estimated trajectory are to the trajectory by the IARA's OGM-MCL system. For this, we employed the MAE metric used in global localization (Equation (6)) to compute the average distance between these two set of poses. In addition, we compared the VIBPT performance improvement when using positioning tracking rather than global localization (VIBGL) only.

The results of these experiments are shown in Figure 5.5 as box-plots having mean, inter-quartile range and whiskers of the error distribution for VIBPT and VIBGL using the 1-meter spacing dataset. Box-plots are shown for the setup UFES-2012 as training and UFES-2014 for test.

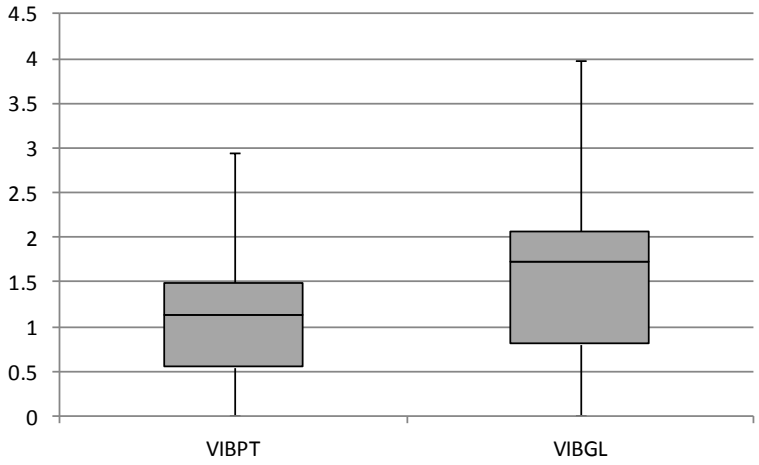


Figure 5.5 - Comparison between VIBPT's Positioning Error and VIBGL's Positioning Error.

As expected the VIBPT's positioning error is smaller than VIBGL's positioning error. Due the EKF's correction step (landmark matches) there is a reduction of about 60 centimeters in the VIBML average positioning error. Furthermore, the positioning error distribution turns more sharp, where 75% of the VIBPT's poses have a positioning error bellow 1.5 meters.

5.2.2 Localization Noise

In order to show the equivalence between the VIBPT and the OGM-MCL system, we evaluated the localization noise of the VIBPT subsystem of VIBML and IARA's OGM-MCL system. For this, we run a set of experiments using the same dataset (UFES-2012) with 1-meter spacing between images, for mapping and localization.

Firstly, we recorded the IARA's OGM-MCL estimated poses by running the OGM-MCL system 10 times along the UFES' campus trajectory and storing the estimated poses, p_i, for each one of the individual laps, L. Then, for each pose $p_{m,i}$ of L_m, we measured the Euclidean distance between $p_{m,i}$ and the corresponding pose, $p_{n,i}$ of lap L_n, for all 10 laps, and calculated the average and standard deviation of these distances. Finally, we used the above mentioned SRPV metric, defined in Equation (7), to compute the mean of these standard deviations. The same steps were followed to compute the VIBPT's localization noise.

5.2.2.1 IARA's OGM-MCL Noise

Figure 5.6 shows, for the IARA's OGM-MCL system, the average of the Euclidean distance between each pose $p_{m,i}$ of lap L_m and the corresponding pose $p_{n,i}$ of lap L_n, for all 10 laps.

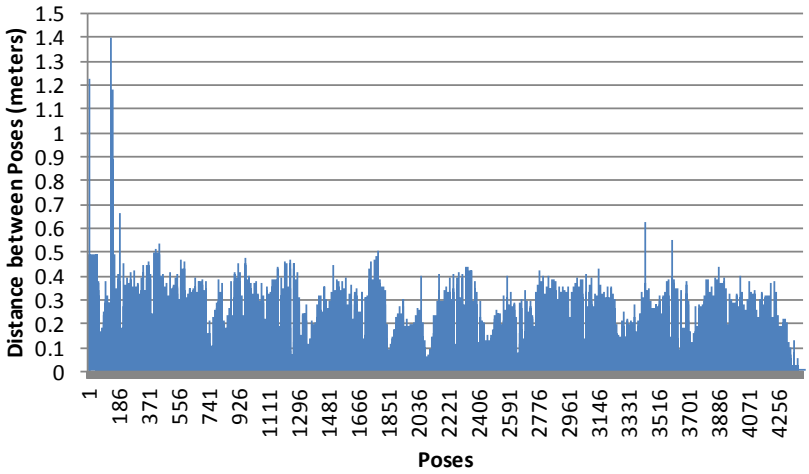


Figure 5.6 - IARA's OGM-MCL localization noise using UFES-2012 dataset for mapping and localization.

In the Figure 5.6, the horizontal axis represents the order, i, of the poses

estimated by the IARA's OGM-MCL system along the UFES' campus trajectory, while the vertical axis represents the average of the Euclidean distances. As the graph in Figure 5.6 shows, except for a few poses, the poses estimated in each one of the 10 laps are very close (less than 1m distance).

To summarize the results show in Figure 5.6 we used the SRPV metric (Equation (7)). We found that the localization noise (mean of the standard deviations) of the IARA's OGM-MCL system is about 0.16m. It is important to note that the resolution of the grid-map of IARA's OGM-MCL is 0.2m. So, a SRPV of 0.16m highlights the good precision of this system.

5.2.2.2 VIBPT Noise

Figure 5.7 shows, for the VIBPT system, the average of the Euclidean distance between each pose $p_{m,i}$ of lap L_m and the corresponding pose $p_{n,i}$ of lap L_n, for all 10 laps.

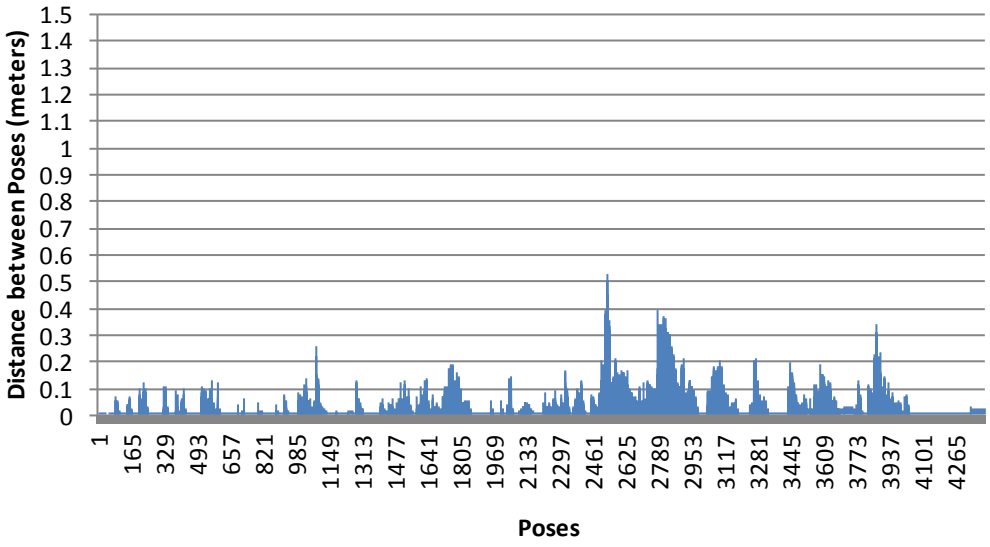


Figure 5.7 - VIBPT's localization noise using UFES-2012 dataset for mapping and localization.

In the Figure 5.7, the horizontal axis represents the order, *i*, of the poses estimated by the VIBPT subsystem along the UFES' campus trajectory, while the vertical axis represents the average of the Euclidean distances. As the graph in Figure 5.7 shows, except for a few poses, the poses estimated in each one of the 10 laps are very close (less than 0.5m distance).

To summarize the results shown in Figure 5.7 we used the SRPV metric

(Equation (7)). We found that the localization noise (mean of the standard deviations) of the VIBPT system is about 0.07m.

Comparing the graphs in Figure 5.6 and Figure 5.7, we can see that the localization noise relative to the VIBPT subsystem is considerably smaller than the noise relative to the IARA's OGM-MCL system.

Although the EKF, used in VIBPT, and the Particle Filter used in the IARA's OGM-MCL system are comparable algorithms, the particle filter has a worse performance when used with a number of particles lower than or close to 1000 units [MAN08]. In the present case, this would explain the higher noise regarding the IARA's OGM-MCL system, since its implementation uses only 1000 particles units.

5.2.3 Localization Displacement

The localization displacement regarding the VIBPT subsystem and the IARA's OGM-MCL system was evaluated by running two localization experiments relative to each one of the systems.

In order to perform these experiments, we firstly built two preliminary maps using the 1-meter spacing UFES-2012 dataset: an occupancy grid map for the IARA's OGM-MCL system, and a Neural Map for the VIBPT subsystem. Subsequently, we test both of these systems using the mentioned maps on the UFES-2012 dataset in the first experiment, and on the UFES-2014 dataset in the second experiment.

Finally, we computed the localization displacement by measuring the MAE of the Euclidean distance (Equation (6)) between the estimated trajectories in the two experiments, for both systems.

5.2.3.1 IARA's OGM-MCL Localization Displacement

Figure 5.8 shows the localization displacement result for IARA's OGM-MCL module. In Figure 5.8 the horizontal axis represents the order of the poses along the UFES' campus trajectory, while the vertical axis represents the Euclidean distance between the estimated trajectories, in meters. Each column (in blue) represents the Euclidean distance between the UFES-2012

and UFES-2014 trajectory's poses, estimated by the IARA's OGM-MCL system. To summarize the results shown in Figure 5.8 we used the MAE metric (Equation (6)). We found that the localization displacement (mean of the Euclidean distances) of the IARA's OGM-MCL system is about 2.40m.

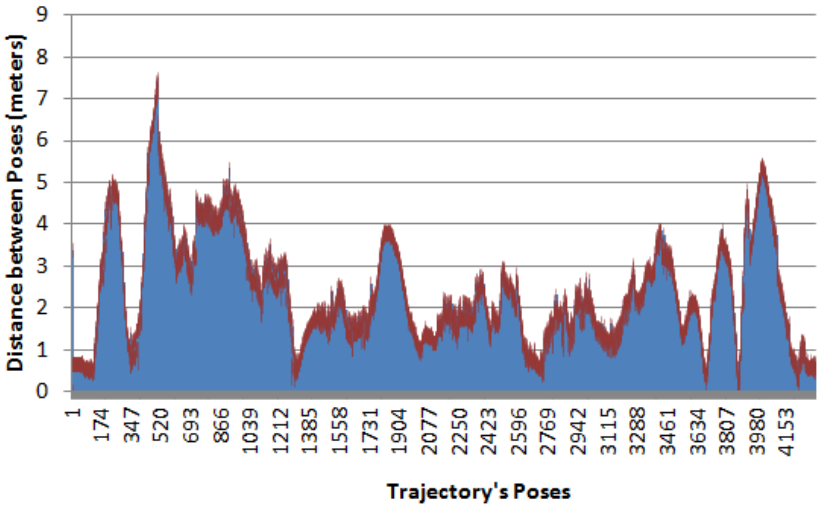


Figure 5.8 - IARA's MCL Localization Displacement. Distance between UFES-2014 and UFES-2012 trajectory's poses are in blue columns. The localization noise regarding IARA's MCL is plotted as error bars (in red).

5.2.3.2 VIBPT Localization Displacement

Figure 5.9 shows the localization displacement result for the VIBPT subsystem. In Figure 5.9 the horizontal axis represents the order of the poses along the UFES' campus trajectory. The vertical axis represents the Euclidean distance between the estimated trajectories, in meters. Each column (in blue) represents the Euclidean distance between the UFES-2012 and UFES-2014 trajectory's poses, estimated by the VIBPT subsystem.

To summarize the results shown in Figure 5.9 we used the MAE metric (Equation (6)). We found that the localization displacement (mean of the Euclidean distances) of the IARA's OGM-MCL system is about 2.61m.

Comparing the graphs in Figure 5.8 and in Figure 5.9, as well as the MAE of both systems, it is possible to observe that the two systems are equivalents. As can be seen in both graphs, the curves are exactly the same for almost the whole trajectory, except for the section of poses from 1385 to 2077, where the VIBPT system have a poor performance compared to the IARA's OGM-MCL

system. This is explained by the fact that in this log's section, the VIBGL subsystem outputs bad global pose estimates sequentially (see Figure 5.10), which causes the VIBPT subsystem to treat them as outliers. Without a reliable global pose and 3D landmarks during so much time, the VIBPT subsystem needs to update its robot's pose estimate using only the visual odometry input. But, once the visual odometry drifts over time the VIBPT's pose estimates has a big error.

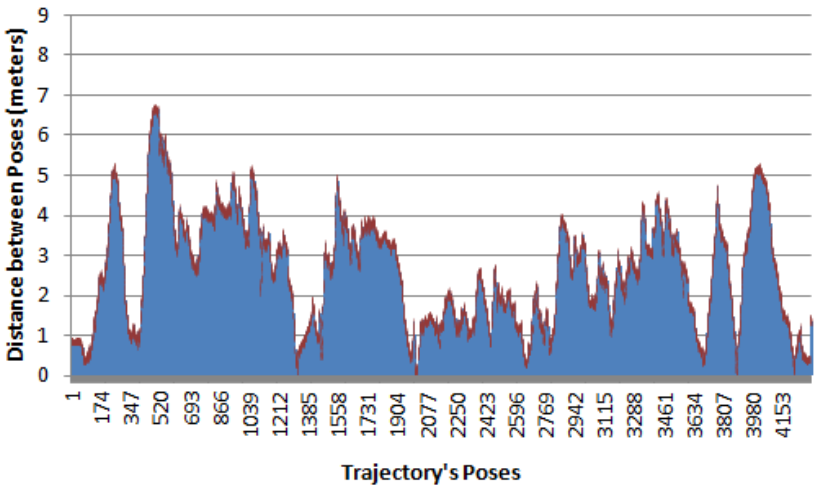


Figure 5.9 - VIBPT Localization Displacement. Distance between UFES-2014 and UFES-2012 trajectory's poses are in blue columns. The localization noise regarding VIBPT subsystem is plotted as error bars (in red).

Although the localization displacement of the system VIBPT is greater than the IARA's OGM-MCL system, it is only about 1.31 standard deviations far away of the OGM-MCL system mean.

Figure 5.10. Samples of the VIBGL's output global pose for poses from 1385 to 2077.

Chapter 6

Discussion

Our results have shown that the VIBML system, purely based on camera images, is able of localizing robots on large maps. Our system was able to map an area of about 3.57km around the UFES' campus, and then locate the IARA robotic platform in this map, with a mean difference of 0.2m when compared to the OGM-MCL approach currently employed in IARA.

As our system uses only images, it does not need any external device, such a GPS, to work. It can be easily adapted to GPS-denied applications and integrated with systems like Google StreetView (where positioning information about the captured images is provided) to perform continuous global localization in Google Maps without the need of communication networks or GPS data.

Many approaches have been proposed to solve the mapping and localization problems using images, as the VIBML subsystem do [MIL08, GLO10, SIV03, MIL12]; however, some of these approaches do not cover continuous global localization and position tracking simultaneously, and don't work with a single input image as the VIBML works.

Our system efficiently solves the problems of mobile robot mapping, global localization and position tracking using only camera images. In a brief

analogy to human beings, the VIBML system has the skills to learn about a certain area (mapping); recognizes a previously learned place by consulting its memories (global localization); and once at a well-known place, localizes itself while navigates through the environment (position tracking).

6.1 Critical Assessment of this Research Work

In this Section, we discuss some of the main shortcomings of our work, focusing on: (i) unreliable initialization, (ii) the kidnapped robot problem, and (iii) the time performance of our system as a whole.

6.1.1 Unreliable Initialization

As shown in Section 5.1, the VIBGL subsystem of VIBML is a good global localizer. It can perform global localization with classification accuracy of about 95% (Figure 5.1) and positioning error smaller than 1.8m (Figure 5.2). However, although these numbers speak in favour of VIBGL, the accuracy of 95% results in unreliable initialization.

When an image is presented to VIBGL, it examines its VG-RAM WNN's memories and returns the pose estimate based on the memory that best fits the input image. This first pose estimate is sent to the VIBPT subsystem of VIBML to initialize the EKF – there is no special treatment to check whether this first pose estimate is correct or if it is a false positive. This can cause initialization failures in some situations. In such cases, the system may believe it is in a certain place whereas, in fact, it is in a completely different place. We have not observed any such situation in our experiments, however.

This problem was not treated in this work, but it can be resolved in many ways. For instance, we could wait for the first five VIBGL's estimations, and choose the robot's pose based on the average between the poses outputted by the VIBGL subsystem.

6.1.2 The Kidnapped Robot Problem

Even though VIBML can perform continuous global localization, it cannot reliably solve the Kidnapped Robot Problem [THR05].

Once the robot is properly localized, VIBML interprets that the global pose estimates computed by the VIBGL subsystem are close to the VIBPT subsystem estimates (see Section 3.4.5). If not, the global pose estimates are treated as outliers and the system remains with its estimative about the last robot's pose. So, in a hypothetical kidnapped condition, when the robot is moved to another place, the VIBML system does not know how to differentiate this situation from a VIBGL outlier pose estimate. However, in the context of our problem of interest (autonomous cars), a kidnapped robot situation is very unlikely and, we believe, does not need to be handled by the VIBML system.

6.1.3 VG-RAM WNN Time Performance

Although the VIBML subsystem can resolve the problems of mapping, global localization and position tracking, and has been shown to be a comparable localization system to another one in the literature, it is not suitable to real-time usage.

When analysing the time performance of the overall system, we identify that the system's modules that consumes most of the system's resources are the VG-RAM WNNs. Specifically, the implementation of such neural networks were made using inefficient filters for translation, scale and Gaussian blurring, that spend most of the computational time.

Chapter 7

Conclusions

In this chapter, we present a brief summary of this work, our conclusions and directions for future work.

7.1 Summary

In this work, we presented and evaluated a novel image-based mapping and localization approach that employs VG-RAM WNNs, dubbed VG-RAM Image-Based Mapping and Localization (VIBML).

We start discussing relevant related works, comparing their advantages and shortcomings with respect to VIBML. We show that, different than other approaches, our system is able to learn about a place with a single image as input and to perform continuous global localization. We, then, presented the subsystems of VIBML: (i) VG-RAM Image-Based Mapping (VIBM), (ii) VG-RAM Image-Based Global Localization (VIBGL), and (iii) VG-RAM Image-Based Position Tracking (VIBPT). Finally, to show that our system solves the problems of mobile robot mapping, global localization and position tracking, we performed a set of experiments regarding the global localization and position tracking mechanisms, and compared them to the Occupancy Grid Mapping and Monte Carlo Localization (OGM-MCL) approach used in our autonomous vehicle, IARA. Our experimental results show that VIBML is

equivalent to IARA's OGM-MCL system.

7.2 Conclusions

Our findings show that the VIBML system is able to perform mapping, global localization and position tracking using only cameras images with performance comparable to the OGM-MCL approach employed in IARA, which uses LIDARs and grid-maps for localization.

In the mapping phase, VIBML receives images of the environment, the positions where they were captured, as well as characteristic points belonging to these images. Subsequently, it learns associations between the images, positions and the images' characteristic points, and uses them as a map of the environment. In the localization phase, VIBML receives images of the environment and uses its previously acquired knowledge – "the map" – to output the positions and the characteristic points representing the places the system believes these images were captured. Finally, it uses the position and the characteristic points to perform global localization and position tracking.

We have tested VIBML in a set of mapping and localization experiments using real-world datasets. Our results show that our system, purely based on camera images, is capable of localizing robots on large scale maps (several kilometers in length) – our system was able to localize an autonomous car in a circuit of 3.57km around the Universidade Federal do Espírito Santo, with a mean difference to the OGMMCL approach of 0.2m. In addition, VIBML was able to localize our autonomous car with average positioning error of 1.12m and with 75% of the poses with error below 1.5m.

7.3 Future Work

The VIBML system opens several avenues of future research. In the near future, we plan to investigate the shortcomings of our system and to extend its functionalities to perform localization in widely used image-maps, like the Google Street View.

One of the shortcomings of the current version of VIBML is unreliable initialization, i.e., it may believe at start up that it is in a certain place when in fact it is not. To try and solve this problem we will investigate better mechanisms for global localization initialization, based on the fact that the VIBGL subsystem of VIBML cannot guarantee its output is bounded by the surroundings of the real robot's position at initialization time.

Another shortcoming is poor performance in terms of time. To overcome this, we plan to implement parallel versions of the translation, rotation and Gaussian filters used in our implementation. These filters consume the most of the computational resources and, since they operate on images data structures, they can be easily parallelized using OpenMP or CUDA enabled GPUs.

The Kidnapped Robot Problem cannot be handled by VIBML and is one of its shortcomings as well. Although unlikely to occur in autonomous cars – our main topic of interest –, this problem can frequently occur with indoor robots. So, to extend the range of applications of VIBML, we will investigate in future works mechanisms for solving this VIBML problem.

We also plan to study the possibility of using VIBML as a replacement of GPS systems so that robots deprived of such devices or in gps-denied environments (where there is no GPS signal) can localize themselves using only images. For this, we will study how to train our system to output positioning information from georeferenced images. One example of database of such georeferenced images that we plan to use in this endeavour is that of the Google StreetView application (see Figure 7.1).

Figure 7.1. UFES campus's trajectory image from Google StreetView.

As the Google StreetView database covers most of the roads and cities of the world, we believe it will soon be possible to use the VIBML system and the Google database to localize cell phone devices without the need of GPS data.

Glossary

BFL	*Bayesian Filtering Library*
CARMEN	*CARMEN Robot Navigation Toolkit*
CMU	*Carnegie Mellon University*
DARPA	*Defense Advanced Research Projects Agency*
EKF	*Extended Kalman Filter*
GPS	*Global Positioning System*
IARA	*Intelligent and Autonomous Robotic Automobile*
IMU	*Inertial Measurement Unit*
iNVT	*iLab Neuromorphic Tookit Vision C++ Tool*
KF	*The Kalman Filter*
LCAD	*Laboratório de Computação de Alto Desemplenho*
LIBELAS	*Library for Efficient Large-scale Stereo Matching*
LIBVISO2	*Library for Visual Odometry 2*
LIDAR	*Light Detection and Ranging*
MAE	*Mean Absolute Error*
OGM-MCL	*Occupancy Grid Mapping - Monte Carlo Localization*
SIFT	*Scale Invariant Feature Transform*
SLAM	*Simultaneous Mapping and Localization*

SRPV	*Square Root of the Pooled Variances*
UFES	*Universidade Federal do Espírito Santo*
VG-RAM	*Virtual Generalizing Random Access Memory*
VIBGL	*VG-RAM Image-Based Global Localization*
VIBM	*VG-RAM Image-Based Mapping*
VIBML	*VG-RAM Image-Based Mapping and Localization*
VIBPT	*VG-RAM Image-Based Position Tracking*
WNN	*Weightless Neural Networks*

Bibliography

[BAY06] Bay, H.; Tuytelaars, T.; and Gool, L. V.; 'SURF: Speeded Up Robust Features', in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin Heidelberg, pp. 404–417, 2006.

[BEE04] Beetz, M.; Schmitt, T.; Hanek, R.; Buck, S.; Stulp, F.; Schroeter, D.; and Radig, B;. The AGILO robot soccer team: experience-based learning and probabilistic reasoning in autonomous robot control. *Autonomous Robots.* Vol. 17, pp. 55-77, Jul. 2004.

[BUE07] Buehler, M.; Iagnemma, K. and Singh, S.; 2007. *The 2005 DARPA Grand Challenge: The Great Robot Race* (1st ed.). Springer Publishing Company, Incorporated.

[BUR96] Burgard, W.; Fox, D.; Hennig, D.; and Schmidt, T. Estimating the absolute position of a mobile robot using position probability grids. *AAAI/IAAI,* Vol. 2, 896-901, 1996.

[CUM08] Cummins, M. and Newman, P.. 'FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance', *The International Journal of Robotics Research*, Vol. 27, No. 6, pp. 647–665, Jan. 2008.

[DAR11] *21st Century Essential Guide to DARPA* - Defense Advanced Research Projects Agency*, Doing Business with DARPA, Overview of Mission, Management, Projects, DoD Future Military Technologies and Science*, DARPA, 2011.

[DAV07] Davison, A. J.; Reid, I. D.; Molton, N. D. and Stasse, O.; "MonoSLAM: Real-Time Single Camera SLAM," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , Vol.29, No.6, pp.1052,1067, June

2007.

[DEL94] De la Escalera, A; Moreno, L.; Puente, E. A; Salichs, M.A, "Neural traffic sign recognition for autonomous vehicles," *Industrial Electronics, Control and Instrumentation, 1994. IECON '94., 20th International Conference on* , Vol.2, No., pp.841,846 5-9 Sep. 1994.

[DIS01] Dissanayake, G.; Newman, P.; Clark, S.; Durrant-Whyte, H. and Csorba, M.;. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation,* Vol. 17, No. 3, pp. 229-241, 2001.

[GEI10] Geiger, A.; Roser, M. and Urtasun, R.; Efficient large-scale stereo matching. In *Proceedings of the 10th Asian conference on Computer vision* - (ACCV'10), Vol. Part I. Springer-Verlag, pp. 25-38, 2010.

[GEI11] Geiger, A.; Ziegler, J.; and Stiller, C.; 'StereoScan: Dense 3d reconstruction in real-time', in *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 963–968, 2011.

[GLO10] Glover, A. J.; Maddern, W. P.; Milford, M. J.; and Wyeth, G. F.; 'FAB-MAP + RatSLAM: Appearance-based SLAM for multiple times of day', in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3507–3512, 2010.

[HEA10] Headrick, T. C.. *Statistical Simulation: Power Method Polynomials and other Transformations*. Boca Raton, FL: Chapman & Hall/CRC, 2010.

[ITT98] Itti, L.; Koch, C. and Niebur, E.; 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, Nov 1998.

[JOC95] Jochem, T.M.; Pomerleau, D.A; Thorpe, C.E., "Vision-based neural network road and intersection detection and traversal," *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on* , Vol.3, No., pp.344,349, 5-9 Aug 1995.

[KLA01] Klaas, G. BFL: {B}ayesian {F}iltering {L}ibrary, 2001. Avaiable at: http://www.orocos.org/bfl.

[LAT11] Lategahn, H.; Geiger, A. and Kitt, B.; 'Visual SLAM for autonomous ground vehicles', in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1732–1737, 2011.

[LOW99] Lowe, D. G.; 'Object recognition from local scale-invariant features', in *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, Vol. 2, pp. 1150–1157, 1999.

[LUD99] Ludermir, T., Carvalho, A., Braga, A., and Souto M., 'Weightless neural models: A review of current and past works', *Neural Computing Surveys*, Vol. 2, pp. 41–61, 1999.

[LYR14] Lyrio Júnior, L. J. ; Oliveira-Santos, Thiago ; Forechi, A. ; Veronese, L. ; Badue, C. ; De Souza, A. F.; Image-Based Global Localization using VG-RAM Weightless Neural Networks. *The 2014 International Joint Conference on Neural Networks* (IJCNN 2014), 2014.

[MAN08] Manya, A. Particle Filter and Extended Kalman Filter for Nonlinear Estimation: A comparative Study (2008), Available at: http://goo.gl/J5acO7

[MIL08] Milford, M. J. and Wyeth, G. F.; 'Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System', *IEEE Transactions on Robotics*, Vol. 24, No. 5, pp. 1038–1053, 2008.

[MIL12] Milford, M. J.; and Wyeth, G. F.; 'SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights', in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1643–1649, 2012.

[MIT98] Mitchell, R. J.; Bishop, J. M; Box, S. K and Hawker, J. F.; 'Comparison of some methods for processing Grey Level data in weightless networks', in *RAM-based neural networks*, J. Austin, Ed. Singapore: World Scientific Publishing Co Pte Ltd, pp. 66–71, 1998.

[NAV05] Navalpakkam, V. and Itti, L.; Modeling the influence of task on attention, *Vision Research,* Vol. 45, No. 2, pp. 205-231, Jan 2005.

[NIS04] Nister, D.; Naroditsky, O. and Bergen, J.; 'Visual odometry', in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern*

Recognition, 2004. CVPR 2004, Vol. 1, pp. I–652–I–659, 2004.

[ROL02] Roland, A.; Shiman, P.; *Strategic Computing: DARPA and the Quest for Machine Intelligence*. The MIT Press, 2002.

[ROU11] Rouff, C.; Hinchey, M.; *Experience from the DARPA Urban Challenge*. Springer, 2011.

[SIM06] Simon, D. *Optimal State Estimation*. John Wiley & Sons. Hoboken, NJ, 2006.

[SIV03] Sivic, J. and Zisserman, A.; 'Video Google: a text retrieval approach to object matching in videos', in *Ninth IEEE Proceedings of International Conference on Computer Vision*, Vol.2, pp. 1470–1477, 2003.

[SOU08] De Souza, A. F.; Badue, C.; Pedron, F.; Oliveira, E.; Dias, S. S; Oliveira, H. and De Souza, S. F.; 'Face Recognition with VG-RAM Weightless Neural Networks', in *Artificial Neural Networks - ICANN 2008*, V. Kůrková, R. Neruda, and J. Koutník, Eds. Springer Berlin Heidelberg, pp. 951–960, 2008.

[SOU13] De Souza, A. F.; Fontana, C.; Mutz, F. W.; Oliveira, T. A.; Berger, M; Silva, A.F.; Oliveira Neto, J.; Aguiar, E.; Badue, C.; "Traffic Sign Detection with VG-RAM Weightless Neural Networks", *Proceedings of the International Joint Conference on Neural Networks*, pp. 730–738, 2013.

[SSE01] Se, S.; Lowe, D.; and Little, J.; 'Vision-based Mobile Robot Localization And Mapping using Scale-Invariant Features', in *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2051–2058, 2001.

[THR05] Thrun, S.; Burgard, W.; Fox, D.; Probabilistic Robotics. *MIT Press*, 2005.

[WOL02] Wolf, J.; Burgard, W.; and Burkhardt, H.; 'Using an Image Retrieval System for Vision-Based Mobile Robot Localization', in *Image and Video Retrieval*, M. S. Lew, N. Sebe, and J. P. Eakins, Eds. Springer Berlin Heidelberg,, pp. 108–119, 2002.