2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

# Image-Based Mapping, Global Localization and Position Tracking Using VG-RAM Weightless Neural Networks

Lauro J. Lyrio Júnior, Thiago Oliveira-Santos, Claudine Badue, and Alberto Ferreira De Souza

*Abstract*— **Humans can easily memorize images of places and labels (road names, addresses, etc.) associated with them, as well as trajectories defined by sequences of images and corresponding positions. Later, they are able to remember places' labels and relative positions when seeing the same images again. In this work, we present an image-based mapping, global localization and position tracking system based on Virtual Generalizing Random Access Memory (VG-RAM) weightless neural networks, dubbed VIBML. VIBML mimics humans ability of learning about a place and of recognizing the same place in a later moment, as well as of tracking self-movement through the environment using images. We evaluated the performance of VIBML on the precise localization of an autonomous car using real-world datasets. Our experimental results showed that VIBML is able to localize car-like robots on large maps of real world environments with accuracy equivalent to that of state-of-the-art methods – VIBML is able to localize an autonomous car with average positioning error of 1.12m and with 75% of the poses with error below 1.5m in a 3.75km path around the main campus of the Federal University of Espírito Santo.**

## I. Introduction

We, humans, can easily memorize images of places where we have been and labels associated with them, such as road names, addresses, etc. We can also remember the relative positions between places when seeing their images again, and track our self-movement through the environment only using images. In this work, we present a novel approach to image-based mapping, global localization and position tracking based on Virtual Generalizing Random Access Memory (VG-RAM) weightless neural networks (VG-RAM, for short) [1], dubbed VIBML (**V**G-RAM **I**mage-**B**ased **M**apping and **L**ocalization, Fig. 1). VIBML mimics humans' abilities of learning about a place and of recognizing the same place in a later moment. It also mimics humans' ability of tracking self-movement through the environment using images.

VIBML solves the mapping, global localization and position tracking problems using camera images only. In robotics, the mapping problem consists in creating a representation of the environment, while the localization problem consists in determining the robot's position with respect to the map [2]. The localization problem can be divided in global localization, position tracking and the kidnapped robot problem (not treated here) [2]. In the global localization problem, the initial robot's position is unknown; i.e. the robot is initially somewhere in its environment, but it lacks knowledge of where it is. In the position tracking problem, the robot keeps track of its position over time, once the initial robot's position is determined.

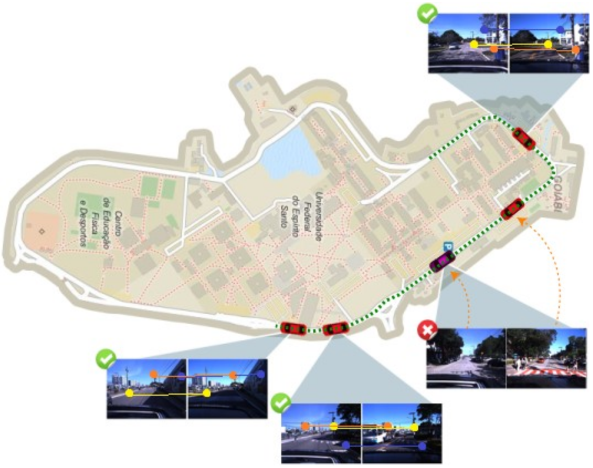

Centro de Educação Física e Desportos
Universidade Federal do Espírito Santo
GOIABI

Fig. 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).

As we have already shown in [3], it is possible to perform global localization using camera images only. However, such approach has a positioning accuracy limited to the global position of the robot in the moment of map construction. In this work, we present an extended system (VIBML) that is able, not only to localize the robot globally, but also to correct its position according to the current point of view of the robot.

The system was extended to store three-dimensional (3D) landmarks along a learned cockpit view trajectory during the map construction. Robot trajectory is assumed to be restricted to the acquired cockpit view, and therefore it must always follow the same orientation of map construction. VIBML performs position tracking by using the learned 3D landmarks (stored in the map) to search for 3D landmarks in currently observed images and uses them to refine the robot's pose. The refinement is achieved by employing an Extended Kalman Filter (EKF), which predicts the robot state based on a car-like motion model and corrects it using a

Lauro J. Lyrio Júnior, Thiago Oliveira-Santos, Claudine Badue and Alberto Ferreira De Souza are with the Departamento de Informática, Universidade Federal do Espírito Santo, Vitoria, ES, 29075-910, Brazil (phone: +55-27-4009-2138; fax: +55-27-4009-5848; e-mail: alberto@lcad.inf.ufes.br).

This work was supported in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq, Brazil (grants 312786/2013-1, 552630/2011-0, and 314485/2009-0) and Fundação de Amparo à Pesquisa do Espírito Santo – FAPES, Brazil (grant 48511579/2009).

978-1-4799-6923-4/15/$31.00 ©2015 IEEE

landmark measurement model.

This paper presents two major contributions: firstly, it describes a system of fundamental importance for autonomous robotic systems that don't have access to GPS signal, but do have access to pre-labeled images with coordinates information such as Google Street View; secondly, it integrates several well known methods in one novel system for image based mapping and localization.

We evaluated the performance of VIBML on the localization of an autonomous car using a set of mapping and localization experiments with real-world datasets. These datasets consist of data acquired from several sensors during laps (a 3.57 km long circuit) performed by our autonomous car around the campus of the Universidade Federal do Espírito Santo – UFES (Federal University of Espírito Santo, Brazil). Our experimental results showed that VIBML is able to localize robots on large maps of real world environments with accuracy equivalent to state of the art methods, like Occupancy Grid Mapping (OGM) with Monte Carlo Localization (MCL) [2]. VIBML was able to localize our autonomous car with average positioning error of 1.12 m and with 75% of the poses with error below 1.5 m.

## II. Background

Much of the work in robotics vision in the last decade relied in visual features with certain degree of invariance to affine transformations [4][5] to provide robust landmarks for mapping and localization [6][7]. Se et al. [6], for instance, developed a vision-based SLAM algorithm for indoor mobile robots using stereo and Scale-Invariant Feature Transform (SIFT) [4]. Both approaches [6][7] (and many similar ones) are mainly conceived as map-based indoor localization and may not be suitable for large outdoor environments.

Several other works focused in situations in which only the initial position of the robot is given. In the seminal work of Nister et al. [8], visual features present in pairs of consecutive video frames are matched so that the camera motion can be estimated from the feature tracks, in a technique called visual odometry. Geiger et al. [9] developed a system where a sparse feature matcher was used in conjunction with a visual odometry algorithm for generating maps of consistent 3D point-clouds. In spite of their capabilities for visual odometry and/or map construction, none of these techniques are suitable for global localization.

In more recent work, Cummins et al. [10] presented FAB-MAP, which is an appearance-based SLAM similar to VIBML, since it allows for continuous global localization by retrieving a previously learned image from the current image. In SeqSLAM [11], Milford et al. described a state-of-the-art appearance-based SLAM that estimates the best matching candidate within a segment of a sequence of previously seen images from a given image. This approach can handle extreme conditions of environment appearance, even for long running distances. Recently, Majdik et al. [12] developed a system that globally localizes a micro aerial vehicle in urban environments using images captured by a single onboard camera. A histogram-voting scheme is used to match those images with georeferenced data provided by Google Street View. In spite of their capabilities for global localization, none of these techniques deal with the position tracking problem.

## III. VG-RAM Image-Based Mapping and Localization (VIBML)

VIBML is an extension of the image-based global localization approach based on VG-RAM that we presented previously in [3]. VIBML integrates global localization and position tracking into a single solution to provide smooth and reliable trajectory estimation. VIBML comprises three main subsystems (Fig. 2): VG-RAM Image-Based Mapping (VIBM), VG-RAM Image-Based Global Localization (VIBGL), and VG-RAM Image-Based Position Tracking (VIBPT).

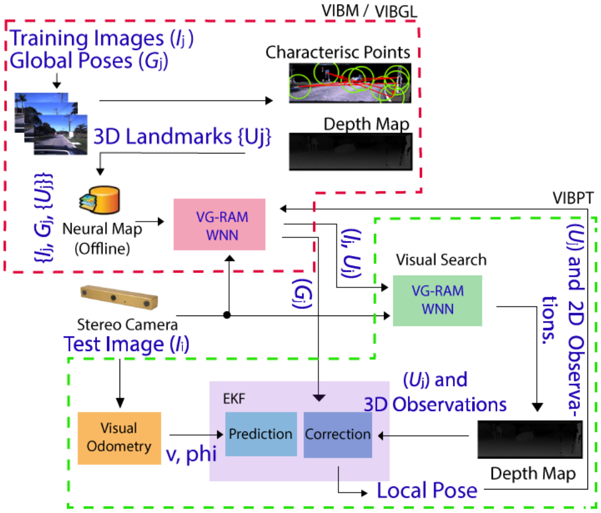


Fig. 2. VIBML system architecture. The VIBM subsystem (red contour) is responsible for mapping (it uses a VG-RAM to learn triples consisted of images, global poses and 3D landmarks sets which represents a place). The VIBGL subsystem (also in red contour) is responsible for the system start up and continuous global localization (it basically recoveries global poses from VIBM's VG-RAM) . The VIBPT subsystem (green contour) is responsible for correcting the global pose estimation and for keeping track of new poses over time.

The VIBM subsystem (red contour in Fig. 2) is responsible for creating a map of the environment. VIBM extends the map representation used in [3] to incorporate 3D landmarks detected on images along a map-making trajectory. These 3D landmarks are used for performing position tracking. Firstly, VIBM receives images of the environment, captured by the robot's stereo camera, along with the robot's global poses where the images were captured. Subsequently, it detects characteristics points on the received images and computes their 3D positions (3D landmarks) using the distance information obtained from depth maps. These depth maps are computed by a stereo matching algorithm [9]. Finally, VIBM learns images, associated global poses and landmarks' positions by employing the VG-RAM architecture described in [3]. This VG-RAM architecture builds a Neural Map of the environment, which is represented by the contents of the memories of the VIBM's VG-RAM neurons.

The VIBGL subsystem (also in red contour in Fig. 2 - described in [3]) is responsible for the system start up and continuous global localization. It receives images of the environment and uses the previously acquired knowledge – the Neural Map – to retrieve the robot´s global poses and the associated landmarks' positions referent to the place where the images were captured during the map construction.

The VIBPT subsystem (green contour in Fig. 2) is responsible for correcting the VIBGL's estimates of the robot's global pose, and for keeping track of the precise poses of the robot over time. It employs an EKF [13] to integrate sensor readings over time through consecutive steps of state prediction and correction.

The state prediction step is performed by means of our robot's motion model, which obeys Ackermann kinematics [14]. The robot's motion model uses velocity and steering angle information, computed from images using visual odometry [9], to estimate the robot pose over time.

The state correction step is performed by means of a measurement model and operates in two phases. In the first phase, VIBPT corrects the robot's local pose by fusing the predicted local pose with the global pose estimated by VIBGL. This procedure ensures that the local pose error is bounded by the global pose error. In the second phase, VIBPT receives the current image of the environment and queries VIBGL for the most similar image in the Neural Map. The 3D landmarks associated with the retrieved image are projected by VIBPT to the camera's coordinate system. Subsequently, VIBPT searches for the projected landmarks in the current image of the environment using a visual search mechanism based on VG-RAM [15]. Once the correspondences are found, VIBPT computes their 3D positions (3D observations) using a depth map computed by a stereo matching algorithm [9]. Finally, VIBPT corrects the robot's pose using the 3D observations as reference (more details in Section III.D)

### *A. VG-RAM Weightless Neural Network*

The VG-RAM is a very effective machine learning technique that offers easy implementation and fast training procedure [1]. 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 each neuron of a neural layer has a set of weightless synapses $S = \{s_1, ..., s_p\}$. The data read from the input layer through the synapses are transformed in a vector of bits $I = \{i_1, ..., ip\}$ (one bit per synapse) using a synapse mapping function that transforms non-binary values from the input layer in binary values.

During VG-RAM training, an input pattern $j$ and its expected output label $t_j$ are set in the input layer and the output of the VG-RAM neural layer respectively. Firstly, each neuron extracts a binary input vector $I_j$ from the input layer, via its set of synapses $S$ (one bit per synapse). Secondly, the expected output label $t_j$ is set in the output of the corresponding neuron in the neural layer. Finally, this input-output pair $L_j = (I_j, t_j)$ is subsequently stored into the neuron's memory, which works as a look-up table.

During VG-RAM 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 memory, the input $I_j$, belonging to the learned input-output pairs $L_j = (I_j, t_j)$ that is the closest (using hamming distance) 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.

For a better explanation of the VG-RAM applied to image classification please refer to [3], and for basic concepts of VG-RAM please refer to [1].

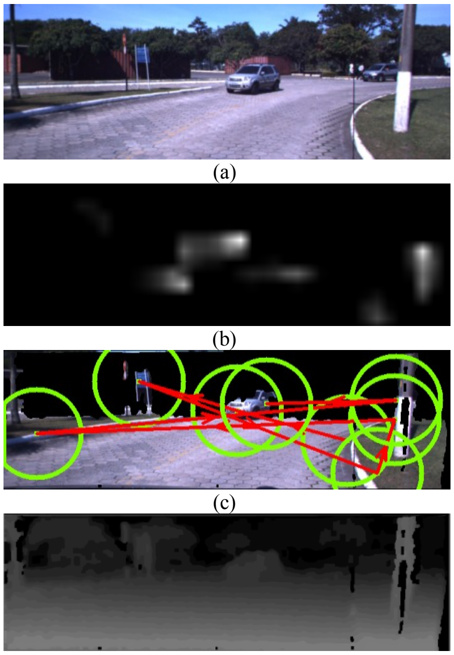
### *B. VG-RAM Image-Based Mapping (VIBM)*

The VIBM subsystem employs a VG-RAM architecture that captures holistic and feature-based aspects of input images by using two different synaptic interconnection patterns [3]. Basically, VIBM learns associations between input images, $I_j$, from the environment, the robot's global poses, and 3D landmarks, that are visible in the images. Let $G_j$ be the robot global pose associated with $I_j$, and $U_j$ be the set of 3D landmarks associated with $I_j$. Let also $\mathbf{T} = \{T_1, ..., T_j, ..., T_{|\mathbf{T}|}\}$ be a set of triplets $T_j = (I_j, G_j, U_j)$ presented to VIBM. In the mapping phase (or training), the $I_j$ 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 $I_j$ with their synapses.

#### *1) Detection of 3D Landmarks*

To select 3D landmarks in the images, the VIBM subsystem employs the iLab Neuromorphic Tookit Vision C++ Tool (iNVT) [16]. We use the iNVT neuromorphic model because it is inspired in the human visual attention, which fits well with our neural based system. This model estimates scene elements (points) that are likely to attract the attention of human observers. These elements are considered the characteristic points or saliencies of an image.

Fig. 3 illustrates the detection of characteristic points on an image. Given an input image (Fig. 3 (a)), the sky is removed, and after the iNVT's visual attention model computes an initial saliency map (Fig. 3 (b)). This saliency map is a combination of color, intensity and orientation feature-maps that are represented as local discontinuities of an image in these modalities. A winner-take-all neural network detects the point of highest contrast in the salience map and draws the focus of attention towards this point (or saliency) (green circles in Fig. 3 (c)). For each shift of attention, an inhibition process in the saliency map (Fig. 3 (b)) is performed to prevent the detection of the same salience twice. After this inhibition process, the saliency map (Fig. 3 (b)) is updated (the next salient point is highlighted) and the above steps are repeated until a certain number of saliencies are selected (all circles of Fig. 3 (c)). Saliencies detected on dynamic objects (like vehicles or pedestrians) should be discarded. Therefore, a threshold proportional to the vehicle's displacement between two image frames is used to drop saliencies that have a relative movement greater than of the vehicle displacement.


(a)

(b)

(c)

(d)

Fig. 3. Detection of 3D landmarks. (a) Scene image. (b) Initial saliency map computed by iNVT. (c) Image saliencies detected by iNVT. (d) Depth map computed by LIBELAS.

To compute the 3D positions of detected saliencies, VIBM employs the Library for Efficient Large-scale Stereo Matching (LIBELAS) [9]. Given a pair of stereo images, LIBELAS computes a depth map (Fig. 3 (d)). By using the information of distance stored in the depth map and the stereo camera's projective parameters, VIBM can compute the 3D positions of the saliencies (i.e. the 3D landmarks that are later used for position tracking).

### C. VG-RAM Image-Based Global Localization (VIBGL)

To perform the global localization, the VIBGL subsystem uses the same VIBM's VG-RAM architecture [3]. As a matter of fact, the VIBGL is only the representation of the VIBM's architecture test phase. Given a query image, VIBGL infers a robot's global pose based on the previously acquired knowledge – the Neural Map. For that, the query image is set as VIBGL's input image and all neurons compute their outputs, which are values of $j$ used during training. Different neurons may vote for different values of $j$, so the most voted value of $j$ is used for determining the output of VIBGL (global pose, $G_j$, and the 3D landmark set $U_j$).

### D. VG-RAM Image-Based Position Tracking (VIBPT)

A major restriction of VIBGL is that it estimates the robot's global pose using previously acquired knowledge – the Neural Map – without performing any correction on the global localization error inherent to the estimated robot's global pose.

In the mapping phase, when VIBGL builds its internal representation of the environment (using the VIBM architecture), it learns that a particular input image, $I_j$, was captured at global pose $G_j$. After that, in the localization phase, when another arbitrary $I_i$, similar to the $I_j$, is presented to VIBGL, it outputs that the inferred image global pose is exactly $G_j$. Nevertheless, this is not necessarily true, since the $I_i$ may have been captured at $G_i$ that is slightly different from the VIBGL's outputted $G_j$ (Fig. 4). This emphasizes the fact that the estimated $G_j$ may usually need to be corrected to best approximate the real $G_i$.

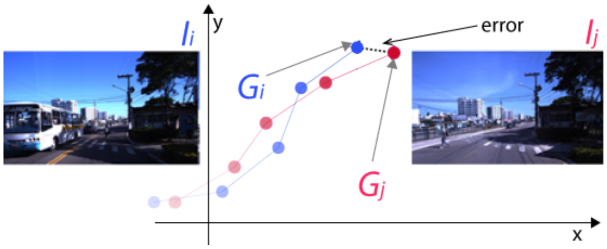


Fig. 4. Error in the global pose of an image estimated by VIBGL. Given a query input image, $I_i$, the global pose, $G_j$, (associated with previously learned $I_j$) estimated by VIBGL may not represent the true $G_i$, because the images may have been acquired from different points of view.

In order to provide a more reliable and precise robot pose, we built the VIBPT subsystem, which integrates the VIBGL's estimated global poses and the matching of 3D landmarks previously stored in the map with 3D observation correspondences. For that, VIBPT employs an EKF [2].

#### 1) Localization with EKF

EKF is a recursive filter that estimates the state of a non-linear system [13]. At a given time, it uses its knowledge about the previous system's state and sensor measurements to estimate the new system's state and the covariance matrix of the estimation error. In this work, we used the EKF in the context of robot localization [2]. The EKF was implemented employing the Bayesian Filtering Library (BFL [17]).

The system state transition model [2] of our EKF implementation was defined by means of the velocity motion model of a car-like robot. This velocity motion model considers the kinematics of a car-like robot [14] and assumes that we can control it through translational velocity ($v$) and steering wheel angle ($\varphi$) commands ($u_t = (v, \varphi)$) computed from visual odometry [9].

The measurement model of our EKF implementation was split in two phases. In the first phase, we use a simple linear measurement model [13] with additive Gaussian noise to fuse the robot's global pose (estimated by VIBGL) with the local pose (estimated by VIBPT in the EKF state prediction step). Thereby, VIBPT can reduce the local pose drift over time and can limit the uncertainty about the local pose within the global pose error. In the second phase, we employ a landmark measurement model [2] that uses the 3D landmarks stored in the Neural Map and their 3D observation correspondences (found by visual search [15]) to update the local pose. For this, the landmark measurement model computes two measurement vectors: *(i)* the expected measurement vector, represented by the distance and angle between the local pose and the 3D landmark pose stored in the Neural Map; and (ii) the observation measurement vector, represented by the distance and angle between the local pose and the 3D observations. Finally, the expected measurement and observation measurement vectors are used by the EKF's landmark measurement model [2] to correct the local poses proportionally to the displacement between them.

### 2) *Visual Odometry*

VIBPT employs the Library for Visual Odometry 2 (LIBVISO2) [9] 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 displacement between two consecutive camera poses $u_t$ can be computed by:

$$v = \frac{\sqrt{\delta_x^2 + \delta_y^2}}{\Delta t} \; and \, \varphi \; = atan2\left(L\frac{\delta_\theta}{\Delta t}, |v|\right), \qquad (1)$$

where $\delta_x$ and $\delta_y$ are the relative displacements in the $x$ and $y$ coordinates, respectively, $\delta_\theta$ is the displacement in the orientation, $L$ is the distance between the front and rear wheels' axles and $\Delta t$ is the time between two image captures.

### 3) *Visual Search of Landmarks*

Fig. 5 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's camera.

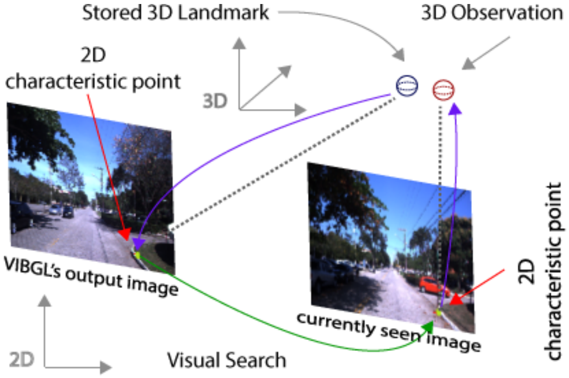


Fig. 5. Visual Search of 3D landmarks (stored in the Neural Map) in the image currently observed by the robot.

Firstly, VIBPT queries VIBGL for the most similar image (left image in Fig. 5), along with its respective 3D landmarks (blue sphere in Fig. 5), to the image currently seen by the robot (right image in Fig. 5). Subsequently, VIBPT reprojects the 3D landmarks outputted by VIBGL back to the camera's coordinate system (left-blue arrow in Fig. 5) in order to obtain the 2D coordinates of the characteristic points. Finally, it searches for these characteristic points in the image seen by the robot (green arrow in Fig. 5). The search (referred as visual search) is performed using a biological inspired detector, also based on VG-RAM [15].

Once the correspondences for each characteristic point are found, VIBPT computes their 3D positions (right-blue arrow in Fig. 5), i.e. the 3D observations represented as a red sphere in Fig. 5, using the depth map computed by the LIBELAS stereo matching algorithm [9].

#### *a) VG-RAM Architecture for Visual Search*

Fig. 6 shows an example of a training instance of our VG-RAM architecture for visual search [15].

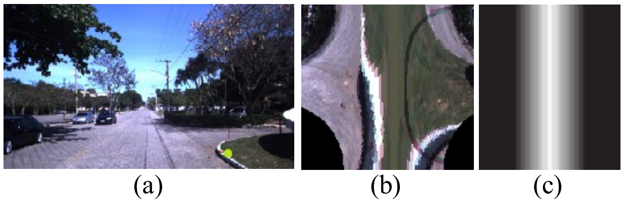


Fig. 6. Example of a training instance of the VG-RAM architecture for visual search. (a) Training image and characteristic point to search for (green dot). (b) Log-Polar centered in the characteristic point of (a). (c) Neurons activation.

In Fig. 6, the network is trained to detect the curb of the street on the image. Fig. 6 (a) shows the training image with the center of attention marked with a green dot; Fig. 6 (b) shows the log-polar mapping of the VG-RAM's input onto the network neural layer; and Fig. 6 (c) shows the output of the neural layer after training. As Fig. 6 (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).

Fig. 7 shows an example of a test instance of our VG-RAM 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. Fig. 7 (a) shows the test image with the found center of attention marked with a green dot and Fig. 7 (b) shows the output of the VG-RAM's neural layer. As Fig. 7 (b) shows, neurons with the centre of their receptive fields over or near the centre of attention generate higher outputs.

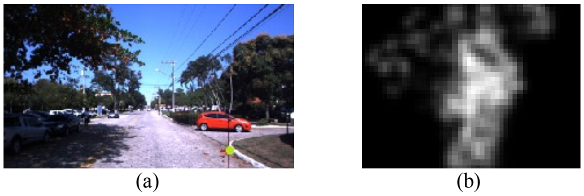


Fig. 7. Example of a test instance of our VG-RAM architecture for visual search.

### 4) *Outlier 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, the Outlier Removal considers the three global pose estimations of VIBGL with higher confidence value to choose the best, $b(g)$, to be used in the linear measurement model. The best pose is defined as the closest global pose to the current local pose, as in Equation ( 2 ):

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

where, $g_x$, $g_y$, $p_x$, and $p_y$ are the position components of the global pose $g$ and the current local pose $p$ respectively.

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, $b(g)$. 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 considered an global localization failure and it is discarded by the system. Note that, the global localization failure is different from the global localization error. The latter is the error attributed to the difference between the current robot pose and the global pose retrieved by the VIBLG (i.e. the pose when the map was constructed).

## IV. Experimental Methodology

### A. Autonomous Robot Platform

We collected the data to evaluate the VIBML system's performance using the Intelligent and Autonomous Robotic Automobile (IARA) [3] developed at the *Laboratório de Computação de Alto Desempenho* – LCAD (High-Performance Computing Laboratory – www.lcad.inf.ufes.br) of UFES.

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 fuses visual odometry pose, Global Positioning System (GPS) pose and Inertial Measurement Unit (IMU) data from IARA's sensors into a precise fused odometry using a Particle Filter [2], and localizes the robot on a previously created occupancy grid map. The OGM-MCL system uses the fused odometry and the robot's motion model (suitable for car-like robots with Ackermann steering [14]) to predict the robot's pose, and correct it by performing a probabilistic matching between the IARA's Velodyne HDL-32 data with the data registered in the grid map.

### B. Datasets

For the experiments, we have used two laps data acquired in different dates. For each lap, IARA was driven with an average speed of about 30 km/h around the UFES campus. A full lap around the campus has an extension of about 3.57 km (Fig. 1). During the laps, image and robot's global pose data were synchronously acquired within 1-meter interval between global poses.

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 include 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.

For a detailed description of the datasets mentioned above, and how the world changed between the two years, please refer to [3]. These datasets are available at http://www.lcad.inf.ufes.br/log.

## V. Experiments

In this section, we show and discuss the outcomes of our experiments. We present the experiments performed to evaluate VIBML in two parts: (i) comparison of positioning error using position tracking plus global localization (VIBPT subsystem) and using global localization only (VIBGL subsystem); and (ii) comparison between VIBML's overall performance and the OGM-MCL system.

The experiments used to evaluate the VIBGL subsystem are thoroughly described in [3].

### A. Positioning Error

To compare the positioning error of VIBPT and VIBGL, we run a set of experiments using the 1-meter spacing UFES-2012 and UFES-2014 datasets [3] for training and testing, respectively.

We measured the positioning error of VIBPT and VIBGL by means of how close their estimated trajectories are to the trajectory estimated by the OGM-MCL system (our ground truth). For this, we measured the Euclidean distance between VIBPT and VIBGL trajectories to the trajectory estimated by the OGM-MCL system. We considered the distance between each pose estimated by VIBPT or VIBGL to the closest pose estimated by OGM-MCL.

Fig. 8 shows the comparison between VIBPT's and VIBGL's positioning error. In Fig. 8, the results are shown as box-plots having mean, inter-quartile range and whiskers of the error distribution for VIBPT and VIBGL.

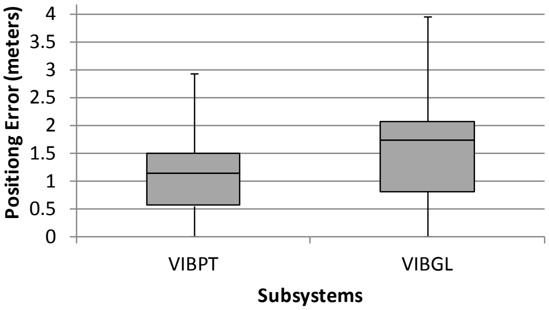


Fig. 8. Comparison between VIBPT's and VIBGL's positioning error.

As shown in Fig. 8, if we use VIBPT, the positioning error decreases of about 0.6m. The reason is that the estimated global pose is corrected by means of the matching of map-stored landmarks with observation correspondences. As consequence, the uncertainty about the robot's pose reduces (the positioning error of more than 75% of the VIBPT's poses are below the VIBGL's average positioning error) and the VIBPT's positioning error becomes smaller than the VIBGL's positioning error.

### *B. Localization Performance*

To show the equivalence between the VIBML system and the OGM-MCL system, we evaluated the localization performance of both systems independently. For this, we measured the localization performance of both systems by means of the localization noise and the localization displacement between the running of the two systems on the UFES-2012 dataset and UFES-2014 dataset.

#### *1) Localization Noise*

To compare the localization noise of VIBML and OGM-MCL, we run a set of experiments using the 1-meter spacing UFES-2012 dataset [3] for mapping and localization.

Firstly, we recorded the OGM-MCL estimated poses by running OGM-MCL 10 times along the UFES' campus trajectory and storing the estimated poses, $p_{m_i}$, for each one of the individual laps, $L_m \in \boldsymbol{L} = \{L_1, L_2, \dots, L_{10}\}$. Subsequently, for each pose $p_{m_i}$ of $L_m \in \boldsymbol{L}$, we measured the Euclidean distance between $p_{m_i}$ and the corresponding pose $p_{n_i}$ of lap $L_n \in \boldsymbol{L}$, $n \neq m$, and calculated the average and standard deviation of these distances. Finally, we calculate the mean of these standard deviations using the Square Root of the Pooled (or weighted) Variances (SRPV [18]) as defined in Equation ( 3 ):

$$SRPV = \sqrt{\frac{1}{q}\sum_{i=1}^{q}\sigma_i^2} \qquad (3)$$

where $q$ is equal to the number of experiments performed and $\sigma_i$ is the standard deviation of the Euclidean distance of the estimated poses between the experiment *i* and *i-1*. The same steps were followed to compute the VIBML's localization noise.

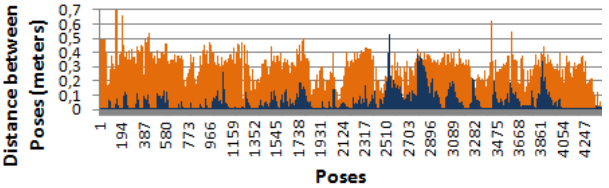


Fig. 9. Localization noise of both systems. OGM-MCL's localization noise is show in orange curve. VIBML's localization noise using UFES-2012 dataset for mapping and localization.

Fig. 9 shows, for OGM-MCL and VIBML, the average of the Euclidean distance in meters between each pose $p_{m_i}$ of lap $L_m$ and the corresponding pose $p_{n_i}$ of lap $L_n$, for all combinations of laps. In Fig. 9, the horizontal axis represents the index, *i*, of the poses estimated by each system along the UFES' campus trajectory, while the vertical axis represents the average of the Euclidean distances. The orange curve represent the OGML-MCL's localization noise while the blue curve represents the VIBML's localization noise.

To summarize the results shown in Fig. 9, the localization noise (mean of the standard deviations) of both systems were calculated using the SRPV metric (Equation ( 3 )). For OGM-MCL the localization noise was about 0.16m. It is important to note that the resolution of the grid-map of OGM-MCL is 0.2m. So, a SRPV of 0.16m highlights the good precision of this system. For VIBML the localization noise was also calculated using the SRPV metric (Equation ( 3 )) and was about 0.07m.

Comparing the curves of both systems in Fig. 9, we can see that the localization noise relative to VIBML is considerably smaller than the noise relative to OGM-MCL. Although the EKF, used in VIBML, and the Particle Filter used in OGM-MCL are comparable algorithms, the particle filter has a worse performance when used with a number of particles lower than or close to 1000 units [19]. In the present case, this would explain the higher noise regarding OGM-MCL, since its implementation uses only 1000 particles units.

#### *2) Localization Displacement*

To compare the localization displacement of VIBML and OGM-MCL, we firstly used the 1-meter spacing UFES-2012 dataset, to build one occupancy grid map, $m_{2012}$, and one neural map, $n_{2012}$, for the OGM-MCL and VIBML systems respectively. Secondly, we used the 1-meter spacing UFES-2014 dataset to build one more occupancy grid map, $m_{2014}$, and one more neural map, $n_{2014}$, for OGM-MCL and VIBML systems respectively.

Subsequently, using the built maps, we tested both of the systems using the UFES-2014 dataset for localization. For this, we ran the OGM-MCL's localization on the $m_{2012}$and $m_{2014}$ maps, and generated two trajectories, $t_{2014}^{o1}$ and $t_{2014}^{o2}$. And after, we run the VIMBL's localization on maps $n_{2012}$ and $n_{2014}$, and generate two trajectories $t_{2014}^{v1}$ and $t_{2014}^{v2}$. Finally, we computed the localization displacement of the systems by measuring the average of the Euclidean distance between the two trajectories ($t_{2014}^{o1}$ and $t_{2014}^{o2}$) estimated by OGM-MCL and the two trajectories ($t_{2014}^{v1}$ and $t_{2014}^{v2}$) estimated by VIBML.

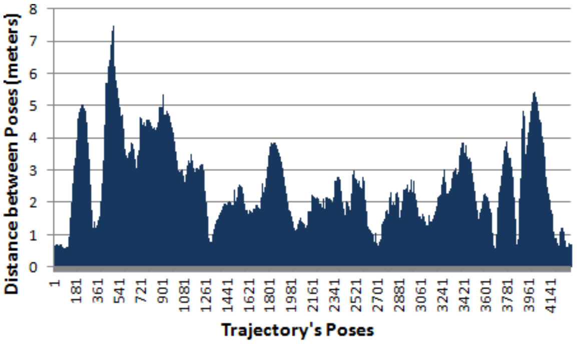


Fig. 11. OGM-MCL's localization displacement.

Fig. 11 shows the localization displacement of OGM-MCL. In Fig. 11, the horizontal axis represents the index, *i*, of the poses along the UFES-2014 trajectory, while the vertical axis represents the Euclidean distance in meters between the two estimated trajectories ($t_{2014}^{o1}$ and $t_{2014}^{o2}$).

To summarize the results shown in Fig. 11, we used the average of the Euclidean distance between the poses. We found that the localization displacement of OGM-MCL is about 2.40m.

Fig. 12 shows the localization displacement of VIBML. In

Fig. 12, the horizontal axis represents the index, $i$, of the poses along the UFES-2014 trajectory, while the vertical axis represents the Euclidean distance in meters between the two estimated trajectories ($t_{2014}^{v1}$ and $t_{2014}^{v2}$).

To summarize the results shown in Fig. 12, we used the average of the Euclidean distance between the poses. We found that the localization displacement of VIBML is about 2.61m.

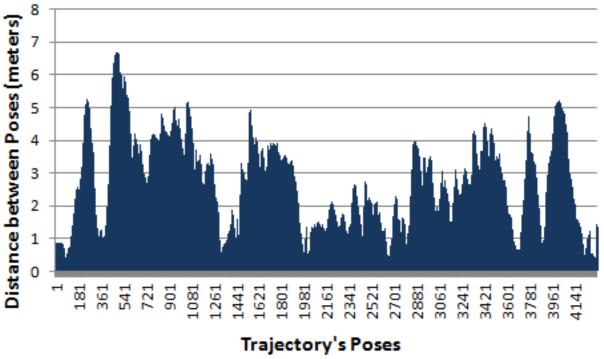


Fig. 12. VIBML's localization displacement.

Comparing the curve signature of graphs in Fig. 11 and in Fig. 12, as well as the average of Euclidean distance of both systems, it is possible to observe that the two systems are equivalents. Although the localization displacements are bigger than 2m in both cases, this is an expected result for this kind of experiment because it focus on behaviour of the displacement over time and not on its absolute value. This is due to the fact that although the vehicle performs the same route in 2012 and 2014, it does not accomplish exactly the same trajectories (i.e. it might have slightly different poses in the same point of the route). As can be seen in both graphs, the curves are mostly the same for the whole trajectory.

## VI. Conclusions and Future Work

In this work, we presented a new image-based mapping, global localization and position tracking approach based on VG-RAM weightless neural networks. After testing VIBML using a set of mapping and localization experiments with real-world datasets, results showed that VIBML is able to localize robots on large maps of real world environments. Our image-based system was able to localize an autonomous car in a circuit of 3.57km with accuracy equivalent to the state of the art OGM-MCL method, which uses LIDARs and grid-maps for localization. VIBML localized our autonomous car with average positioning error of 1.12m and with 75% of the poses below 1.5m error. In addition, the position tracking functionality of VIBML decreased the positioning error of the previous VIBGL's system [3] by 0.6m.

Although VIBML presented a good performance in regards to position tracking, it has shortcomings, including: unreliable initialization, since in 5% of the cases (as showed in [3]) the global localization might fail; and the poor time performance as a whole, since we were interested in a proof of concept and therefore we have not put effort on optimizing the filters used.

Directions for future work include to address the issues raised above, and to extend the VIBML to perform localization in widely used image-maps like Google Street View.

## References

[1] T. Ludermir, A. Carvalho, A. Braga, and M. Souto, "Weightless neural models: A review of current and past works," *Neural Computing Surveys*, vol. 2, pp. 41–61, 1999.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, Mass.: MIT Press, 2005.

[3] L. J. Lyrio Júnior, T. Oliveira-Santos, A. Forechi, L. Veronese, C. Badue, and A. De Souza, "Image-based global localization using VG-RAM Weightless Neural Networks," in *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 3363–3370.

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

[5] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin Heidelberg, 2006, pp. 404–417.

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

[7] J. Wolf, W. Burgard, and H. Burkhardt, "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, 2002, pp. 108–119.

[8] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, vol. 1, pp. 652–659.

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

[10] M. Cummins and P. Newman, "Appearance-only SLAM at Large Scale with FAB-MAP 2.0," *Int. J. Rob. Res.*, vol. 30, no. 9, pp. 1100–1123, Aug. 2011.

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

[12] Y. A.-S. A. L. Majdik, "MAV urban localization from Google street view data," *Proceedings of the ... IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3979–3986, 2013.

[13] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*, 1 edition. Hoboken, N.J: Wiley-Interscience, 2006.

[14] A. Weinstein and K. L. Moore, "Pose estimation of Ackerman steering vehicles for outdoors autonomous navigation," in *2010 IEEE International Conference on Industrial Technology (ICIT)*, 2010, pp. 579–584.

[15] A. F. De Souza, C. Fontana, F. Mutz, T. Alves de Oliveira, M. Berger, A. Forechi, J. de Oliveira Neto, E. de Aguiar, and C. Badue, "Traffic sign detection with VG-RAM weightless neural networks," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 2013, pp. 1–9.

[16] V. Navalpakkam and L. Itti, "Modeling the influence of task on attention," *Vision Research*, vol. 45, no. 2, pp. 205–231, 2005.

[17] Klaas, G, *BFL: {B}ayesian {F}iltering {L}ibrary*. 2001.

[18] T. C. Headrick, *Statistical Simulation: Power Method Polynomials and Other Transformations*. Boca Raton: Chapman and Hall/CRC, 2009.

[19] Manya, A, "Particle Filter and Extended Kalman Filter for Nonlinear Estimation: A comparative Study," Technical Report, 2008.