2017 International Conference on Military Technologies (ICMT)
May 31 – June 2, 2017, Brno, Czech Republic

# Multiple People Detection and Identification System Integrated with a Dynamic Simultaneous Localization and Mapping System for an Autonomous Mobile Robotic Platform

Alberto Torres Angonese*, Paulo Fernando Ferreira Rosa†
*Instituto Militar de Engenharia, Rio de Janeiro, Brazil, e-mail: *angonesealberto@gmail.com*,
†Instituto Militar de Engenharia, Rio de Janeiro, Brazil, e-mail: *rpaulo@ime.eb.br*

*Abstract*— **This paper presents the integration of a multiple people detection and identification system with a dynamic simultaneous localization and mapping system for an autonomous robotic platform. This integration allows the exploration and navigation of the robot considering people identification. The robotic platform consists of a Pioneer 3DX robot equipped with an RGBD camera, a Sick Lms200 sensor laser and a computer using the robot operating system (ROS). The idea is to integrate the people detection and identification system to the simultaneous localization and mapping (SLAM) system of the robot using ROS. The people detection and identification system is performed in two steps. The first one is for detecting multiple people on scene and the other one is for an individual person identification. Both steps are implemented as ROS nodes that works integrated with the SLAM ROS node. The multiple people detection's node uses a manual feature extraction technique based on HOG (Histogram of Oriented Gradients) detectors, implemented using the PCL library (Point Cloud Library) in C ++. The person's identification node is based on a Deep Convolutional Neural Network (CNN) that are implemented using the MatLab MatConvNet library. This step receives the detected people centroid from the previous step and performs the classification of a specific person. After that, the desired person centroid is send to the SLAM node, that consider it during the mapping process. Tests were made objecting the evaluation of accurateness in the people's detection and identification process. It allowed us to evaluate the people detection system during the navigation and exploration of the robot, considering the real time interaction of people recognition in a semi-structured environment.**

*Keywords*— **People Detection, HOG, Deep Learning, CNN, Simultaneous Localization and Mapping (SLAM), robot operating system ROS;**

## I. Introduction

The human and robot interaction is a reality presented nowadays. For performing tasks as search and rescue, urban disasters, hostage situations, domestic services or elderly assistance, it is expected that an autonomous robot is capable of locating itself while exploring the environment. This issue characterizes the research area known as SLAM (Simultaneous Localization and Mapping). For those applications, it is interesting that the robot considers the detection and identification of people while performing SLAM in a dynamic environment and thereby, modifying its decision-making actions based on this detection. This paper presents a people detection and identification method that runs simultaneously with a SLAM method, already implemented and presented in a previously paper [1].

The human recognition has been the focus of research for a long time. Despite the impressive results achieved by a wide range of techniques, it still presents many challenges to overcome, especially considering its application in mobile autonomous robotic. In [2], the authors present a methodology for people detection and tracking from RGBD images extracted by a mobile robotic platform. The paper describes a real time people detection and tracking application based on the Histograms of Oriented Gradients (HOG) 's technique. Notwithstanding positive results presented by the authors, the application is limited to the people detection issue.

Deep Learning techniques have emerged as an interesting alternative to leverage advances in the area of visual recognition and machine learning as can be seen in [3], [4] and [5]. Deep convolutional neural networks have been successfully used in several problems involving image recognition, such as face detection, as described in [6], [7] and [8]. The motivation for using a deep CNN architecture has emerged as a possibility of treating high-level abstractions in a more efficient way, as the case of people identification as illustrated in figure 1. The figure shows the performance of our CNN identifying and labeling two different people.

This article addresses the people detection and identification problem in two steps. The first one is the detection and the second the identification. The first step is an application based on the Point Cloud Library (PCL) [9]. The second step presented in this paper uses the Matlab MatConvNet library [10] to implement a deep Convolutional Neural Network architecture as an improved solution for people recognition. Both approaches are implemented as applications in the robotic operating system, called ROS nodes. They are embedded with the SLAM ROS node in a robotic platform able to perform simultaneous localization an mapping, allowing the robot's exploration and navigation to be made considering the interaction with detected and identified people in a semi-structured environment.

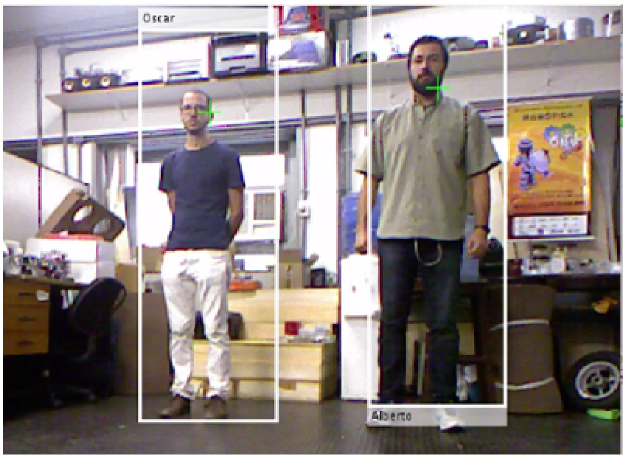


Figure 1. CNN People Identification

Tests were performed in order to evaluate the accurateness of the people identifying approach in the second step. The contribution deals with searching of computational methods enabling robust detection to variations on the position and partial occlusions, allowing its use in unstructured environments.

The majority of researches dealing with SLAM do not take into account the presence of people interacting in the environment which is being mapped by the robot, even with exploration and mapping of multiple cooperatives robots as in [11] or considering dynamics environments as in [12]. People detection and SLAM systems were integrated by robot operating system ROS [13] embedded on the robot. The general idea is to change the robot navigation once human is identified. For this, it was developed specific ROS nodes working simultaneously to SLAM system, which is being currently developed at the laboratory of Military Institute of Engineering.

Besides introduction, this paper is also composed of the additionally sections. In Section II is described the implementation of people detection and identification system. Furthermore, it will also be described comparative tests and results in order to evaluate its accuracy. Section III presents the description of the robot hardware composition and ROS nodes system integration. Section IV presents a real experiment that uses the robotic platform described. Finally, Section V proposes some final considerations.

## II. People Detection

The people detection and identification system was implemented using two distinct approaches. The first one is based on HOG detector and the second one is base on a Deep CNN architecture. Both are implemented as ROS nodes that integrates with SLAM system robot. Figure 2 provides an overview of the system steps. In accordance with figure, as the robot starts exploring the environment, it extracts RGBD images that are processed by the people detection node. The people identification node process the information from the previous node so returning the person's pose $(X_P, Y_P, \theta_P)$ in relation to the robot's pose $(X_R, Y_R, \theta_R)$ and to the map, that is being generated by the SLAM process.

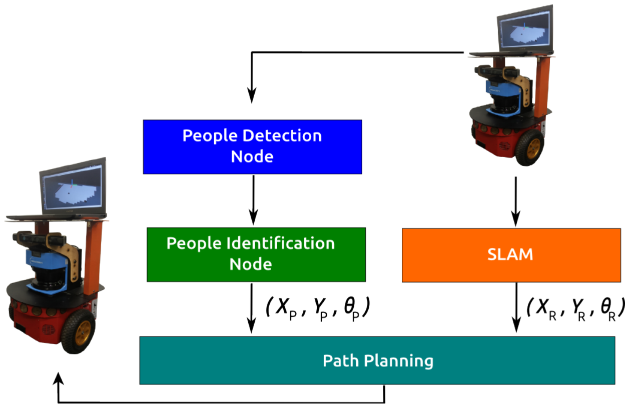


Figure 2. System Overview

### A. People Detection Node

People Detection Node step is based on HOG descriptors that are implemented using the Point Cloud Library (PCL)[14].

The people detection node has been adapted from the Ground Based People Detection App application available on the PCL. The current detection process implemented, uses a labeled dataset from the Kinect Tracking Precision (KTP) Dataset [15] only for SVM classification process. It is important to highlight that the training step was not performed in our implementation.

During the HOG experimentation, it was verified a satisfactory real time multiple people detection, even with partial occlusions in semi-structured indoor environments, as shown in the figure 3. However, the system was not robust to unstructured outdoor environments with many false positives detection. Moreover, another characteristic is that the system presents the detection of people in upright position, with a frequent loss when different positions from the upright which regularly occurs in unstructured environments. This kind of issue was addressed in a previous work as described in [16].

Figure 3. Multiple People Detection

### B. People Identification Node

The identification node is implemented as a Deep Convolutional Neural Network (CNN) architecture using the MATLAB MatConvNet [10] library. That library implements convolutional neural networks for computational vision applications. Presently, there are numerous libraries for implementing deep learning architectures and CNNs as CudaConvNet[17], Theano[18], TensorFlow[19] and Caffe[20].

The MatConvNet was adopted in this work because of its integration with MATLAB. Besides the rapid and easy to use environment, the MatConvNet possesses many pretrained CNN models, that gained prominence in the Imagenet ILSVRC challenge for image classification, as the case of AlexNet proposed in [21], of VGG that implements the described model in [22], of VGG-VD in [23] and GoogLeNet presented in [24].

In this paper it was used a variation of the GoogLeNet CNN architecture provided by the MatConvNet framework. The network has 153 layers structured as follows:

- A linear chain, composed of three Convolutional layers(Conv) with 7x7, 1x1 and 3x3 patches size, four Rectified Linear Unit (ReLU) activation layers, two 3x3 Max Pooling layers, and two Local Response Normalization (LRN) layers. All those are sequentially connected as described in figure 4.
- Nine modules are structured for performing convolutions and poolings at different patches and strides. These modules are the basis of the main network architecture. Each module wraps a set of parallel computational blocks as illustrated in the figure 5. It has one 3x3, one 5x5 and four 1x1 Convolutional Layers, followed by their respective ReLU layers and a 3x3 Max Polling layer. The 1x1 Convolutional Layers are responsible for the convolutional input dimension reduction. This approach enables the network to perform convolutions with larger filter inputs with high performance.
- Two other auxiliary blocks of layers are used, aiming the improvement of the depth representations of the data. Each block is composed of an average polling layer with 5x5 patch size, a 1x1 convolutional with the respective rectified linear activation layer, two full connected layers with 1024 units and the ReLU layers interconnecting them, and a softmax classifier layer.
- The last three layers are similarly disposed of as the auxiliary layers, with a 7x7 average layer, a full connected layer, and the last softmax classifier layer.

The GoogLeNet Convolutional Neural Network architecture was adopted in this paper because of the high-performance results with low computational requirements. In this approach it was used the Imagenet GoogLeNet pre-trained model obtained from MatConvNet [25] to extract features from our own labeled dataset $(person_1, person_2 ..., person_n)$ creating a people CNN Feature model that are used to train a multiclass SVM classifier to identify different people in images. The

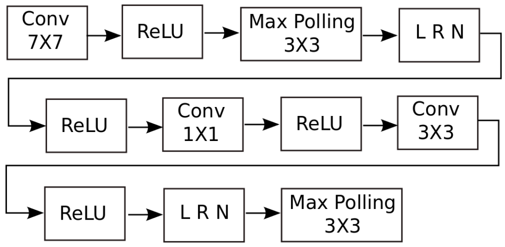


Figure 4. Initial Linear Layers

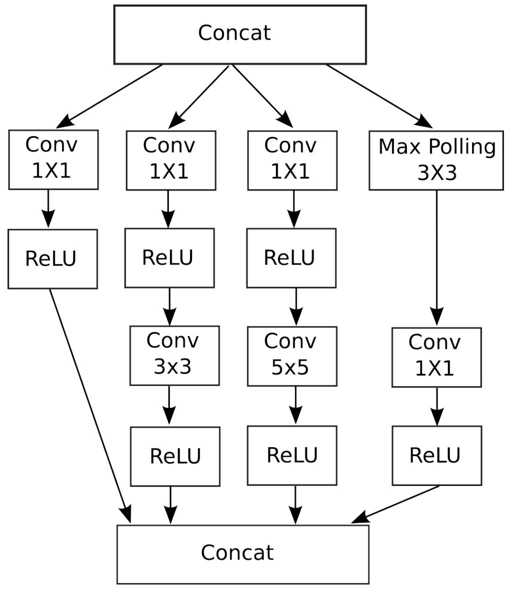


Figure 5. Main Network Module Layers

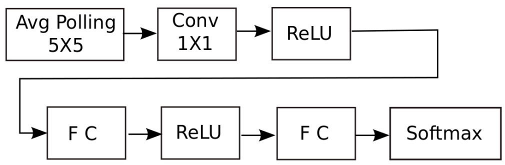


Figure 6. Auxiliary Network Block Layers

advantage of using this method is that a pre-trained CNN architecture has already learned a rich set of features that can be used for a wide range of images. Besides to reduce training time, it can be useful to fine tuning the feature model for specific purposes, such as the case of person identification.

The figure 7 shows an overview of the two nodes working together. In the top of figure 7 is represented the people detection node and in the bottom it is illustrated the flowchart with an overview of the method adopted for the people identification node.

After the correct people detection is performed by the first node, the centroids and the heights of the detected people are published as topics with ROS messages to the next step, for identification. The People Identification Node subscribes the topic from the previous step with the messages containing the centroid and height of the detected people. This information is used to define a Region of Interest (ROI) which is cropped and classified

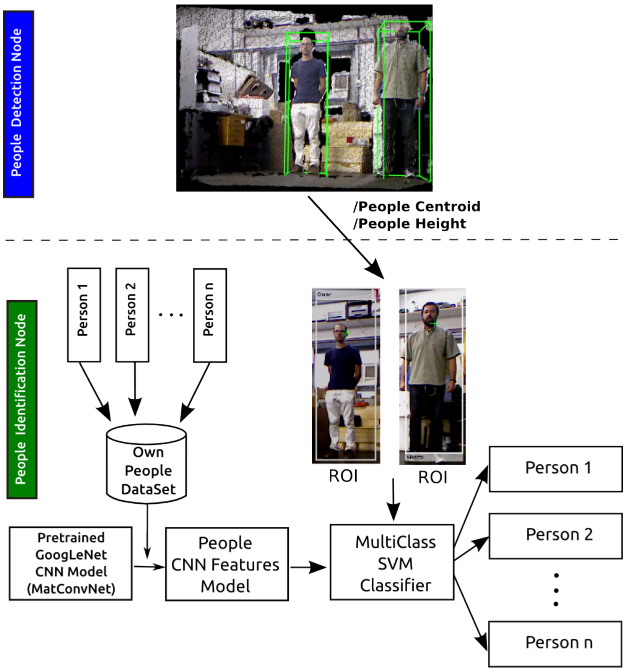


Figure 7. CNN People Identification Overview

by the multiclass SVM, trained from the CNN feature model provided by our own people dataset.

*1) People Identification Accurateness Test:* The test's aim is the accurateness evaluation of the People CNN Feature Model approach for person identification. The classification process was performed with 400 cropped images extracted from an offline video capturing the randomly walking of two people previously labeled in the People CNN Feature Model.


Figure 8. People Identification Test Scenario

As shown in figure 8 the accuracy test, was performed considering two classes of people labeled as *Person 1* and *Person 2*. The CNN architecture was trained with 250 positive images of each class and one more class with 250 negative images, totaling 750 of training images.

The scenario illustrated in figure 8 demonstrates the trained people walking randomly. Table I shown the confusion matrix and table II presents the accurateness results of the identification process.

TABLE I
CONFUSION MATRIX OF PEOPLE IDENTIFICATION NODE

| | Person 1 | Person 2 |
|---|---|---|
| Person 1 | 334 | 66 |
| Person 2 | 10 | 390 |

TABLE II
ACCURATENESS OF PEOPLE IDENTIFICATION NODE

| | Person 1 | Person 2 |
|---|---|---|
| True Positive | 334 | 390 |
| False Positive | 66 | 10 |
| Total Images | 400 | 400 |
| Accurateness | **84%** | **98%** |

### *C. Test and results considerations*

The performed classification test showed satisfactory results in terms of accuracy. Even considering the presented scenario in which many of the cropped images presents poor information for a correct classification as it is shown in figure 9.

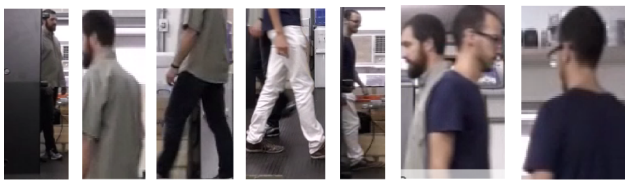

Figure 9. Example of cropped images with mismatch classification

However, when both nodes work together, the frame rate of the detection node must be reduced to 5 fps so that the correct identification by the identification node occurs. Besides, the computational cost to train the CNN feature's model using a deep learning architecture such as the GoogLeNet requires a GPU for satisfactory computing time results, which is easily solved by the tools used in this research such as MatConvNet.

## III. ROBOT SYSTEM

In this section, it is described the robot's platform and the ROS nodes components developed for the integration of the people detection and identification system and the SLAM system.

### *A. Robot Plataform*

The robot illustrated in figure 10 is composed of the described components as it follows:

- Robotic Platform Pioneer 3DX: It is a widely used platform for research related to mobile robots. It is composed of encoders on wheels, a battery and a set of sonar already embedded. Additionally, it enables the angular and linear speeds control.
- Laser Sick LMS200: It is used for the 2D feature extraction during the SLAM process.

- RGB-D Camera: Besides being used in SLAM process, it is the principal component of scenes capture that will be used in people detection.
- Computer: 2.40GHz Intel Core i7 notebook, running a 64-bit Ubuntu 14.04 LTS Linux distribution, with 8 GB of RAM and a 2GB dedicated video board. It is configured with robot operating system ROS, that is the main component for SLAM applications and people detection and identification processing.

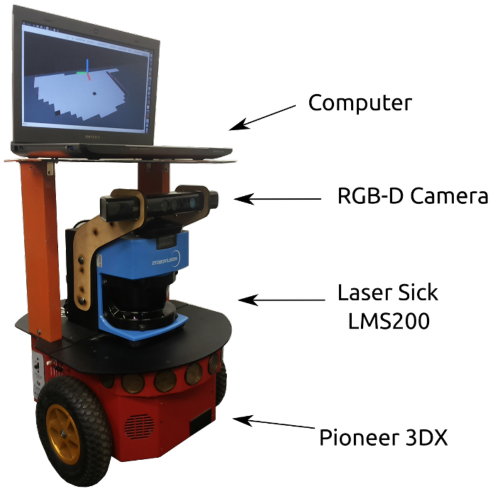


Figure 10. Robotic Platform

### B. People Detection and SLAM ROS Nodes Components Integration

The People Detection and Identification and SLAM applications are integrated by robot operating system ROS. The physical devices as the sensor laser and the RGBD camera provides information to SLAM applications and People Detection trough ROS which implements an efficient exchange messaging system. It allowed us to integrate People Detection and Identification node to SLAM node with the minimum of possible adaptations. The message structure used in this paper is presented in figure 11. The implemented nodes communicate with each other by publishing messages to topics. In the figure 11, the ellipses are the nodes and the rectangles are the representation of the topics. The message types are detailed as follows:

- **/Angle** : This is a standard array of float32 type ROS message. The publisher is the PeopleDetector node, and the subscriber is the PeopleIdent node. It is the people's angle array $(\theta_{P1}...\theta_{Pn})$ between detected people and RGBD camera.
- **/Height**: This is a standard array of float32 type ROS message. The publisher is the PeopleDetector node, and the subscriber is the PeopleIdent node. It is the people's height array $((H_{P1})...(H_{Pn}))$.
- **/Distance**: This is a standard array of float32 type ROS message. The publisher is the PeopleDetector node, and the subscriber is the PeopleIdent node. It is the distance between detected people and RGBD camera.
- **/PeopleCentroid** : The message type is an array of points. The point is an (x, y, z) float32 message type. The publisher is the PeopleDetector node and the subscriber is the PeopleIdent node. It is the centroid array $((X_{P1}, Y_{P1}, Z_{P1}\theta_{P1})...(X_{Pn}, Y_{Pn}, Z_{Pn}\theta_{Pn}))$ of the detected people.
- **/camera/rgb/image_rect_mono**: This is a point cloud library (PCL) RGB image message type, acquired from the RGBD camera. It is simultaneously used by the People Identification node for CNN classification process and by the SLAM node.
- **/camera/depth_registered/points**: This is a PCL point cloud message type, acquired from the RGBD camera. It is used by SLAM node.
- **/CNN_PersonCentroid**: This is a standard ROS point (x, y, z) float32 type message, published by PeaopleIdent node and subscribed by the SLAM node. It is the centroid of the identified person.

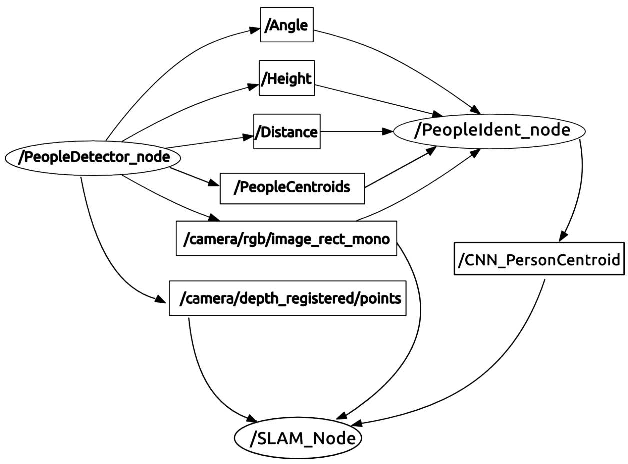


Figure 11. Ros Message Structure

Another interesting characteristic is that ROS implements a hardware abstraction set, providing the possibility of sensors substitution or even the entire robotic platform without any significant impact on applications already developed. It allows us to constantly maintain an updated system, testing more efficient and robust components.

The ROS node components for people detection and SLAM system are illustrated in figure 12. Each of the robot hardware devices has an abstraction on ROS, as showed in the figure. The orange rectangles illustrate the physical parts coupled to the robot. Odometry, Left Motor, and Right Motor are components of the robot pioneer described in this paper. The RGBD Camera and the Laser Sensor are components coupled to the robot and are responsible for optical sensing and features extraction, respectively. The blue ellipses are ROS abstractions related to physical components. The topics, illustrated by the blue rectangles, are responsible for the messages coming from the ROS nodes, specifically the encoders for odometry, laser and camera RGBD, that feeds the SLAM application node and People Detection node. The people detection system uses ROS components that are highlighted by the yellow rectangle, which principal sensor is the RGBD camera, responsible for scene extraction detection. After people detection node processing, a message containing the people's pose array $(X_{P1}, Y_{P1}, Z_{P1}\theta_{P1})...(X_{Pn}, Y_{Pn}, Z_{Pn}\theta_{Pn})$ and the people's height array $(H_{P1})...(H_{Pn})$ is published to the people identification node that subscribes the messages. After that, the people identification topic publish the identified person's pose $(X_{Pi}, Y_{Pi}, \theta_{Pi})$ to the SLAM node. Basically, the SLAM node adjusts the person and robot coordinate system with the map coordinate system as follows:

$$px = Z_{Pi} + D_X RGBD$$

$$py = -Y_{Pi} - D_Y RGBD$$

$$X_P = px \cos\theta_R - px \sin\theta_R + X_R$$

$$Y_P = py \sin\theta_R + py \cos\theta_R + Y_R$$

Where $D_X RGBD$ and $D_Y RGBD$ are the $X$ and $Y$ RGBD camera distances from the center of the robot. Then, the SLAM topics publish the people's pose $(X_P, Y_P, \theta_P)$ to the path planning node that, along with the robot's pose $(X_R, Y_R, \theta_R)$ are processed and returned to Pioneer node for robot move control.

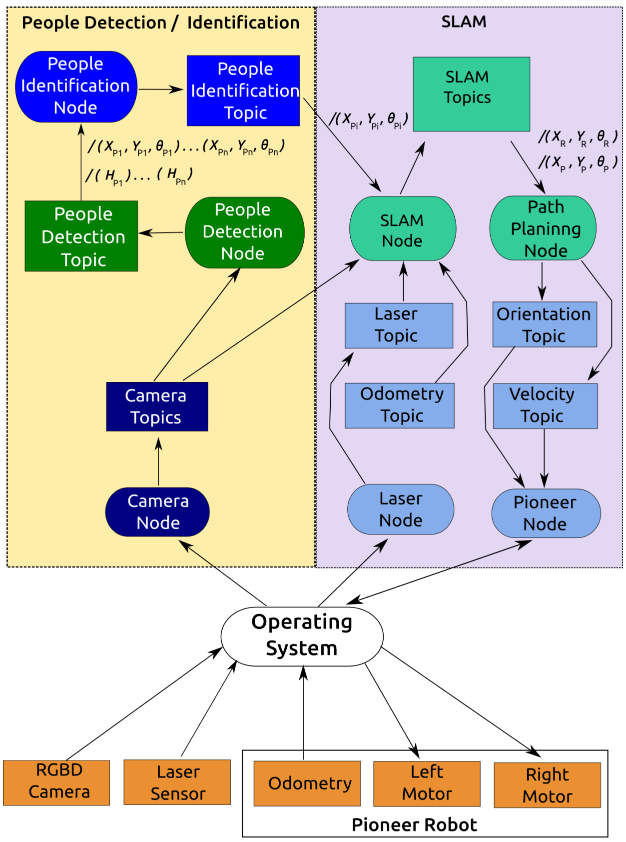


Figure 12. People Identification and SLAM ROS Nodes Components

## IV. Experiments

### A. Follower

As a way to evaluate the integration of detection system with the robot operation in real time, it was implemented a simple follower application in which in case of detecting and identifying a person, the robot turns, and it follows the identified person, maintaining a 50 cm distance. The experiment was conducted at the Military Institute of Engineering facilities in a semi-structured environment. For this test, the RGBD camera's frame rate was configured to 5 fps, and the linear velocity of the robot was setup to 5 cm/s. These values were empirically obtained during the test performing, with the purpose of synchronizing the video acquisition and processing with the robot's moving, ensuring a safe identification. The figure 13 illustrate the robot following the labeled *Person2*. After the identification, the robot follows the *Person 2*, moving from position 1 to 2.


Figure 13. External Image of the Robot Following the Labeled *Person 2*

The figure 14 demonstrates the both nodes performing the detection and identification. The top left image in figure 14, shows the bounding box of the detected people, at the right, it is the people identification node, in which the two detected people are properly identified. The person to be followed are previously defined, and in this experiment are the one labeled as *Person 2*. In the bottom left of the figure 14 it is an external view, showing the robot following the labeled person. The application of the follower robot comprises the following steps:

1 - After positive detection of the people, performed by the people detection node , it is calculated the angle $\theta = \tan^{-1}\left(\frac{y_H - y_R}{x_H - x_R}\right)$ and distance $d = \sqrt{(x_H - x_R)^2 + (y_H - y_R)^2}$ from the center of the person detected in relation to the RGBD sensor coupled to the robot.

2 - The people's centroid array $(X_{P1}, Y_{P1}, Z_{P1}\theta_{P1})...(X_{Pn}, Y_{Pn}, Z_{Pn}\theta_{Pn})$ and the people's height array $(H_{P1})...(H_{Pn})$ are published into the people identification node for the deep CNN classification process.

3 - Posteriorly, it is calculated the robot angular velocity $w = \theta - (\frac{\pi}{2})$. The following step verifies whether the angle $\theta \approx 90$ degrees and distance $d \geq 50$ cm, if yes it is attributed a constant linear velocity of $5cm/s$ on the robot x axis, moving the robot direction towards the detected person. Otherwise, it is attributed a linear velocity $= 0$ and an angular velocity w, turning the robot around $\theta \approx 90$ degrees.

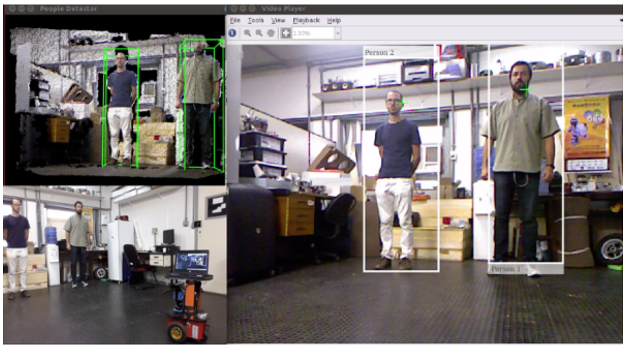

Figure 14. Robot following the labeled *Person2*:In the top left, it is the people detection node. On the right, it is the identification node and in the bottom left it is an external video of the experiment.

The implantation of the follower application presents a simple setting. Besides, it was configured a slow linear and angular velocity for the robot's moving, because of frame rate reducing, as described in section II-C, limiting its application in structured environments. However, it supported to evaluate the detection process in real time running jointly with the SLAM application.

### *B. People Detection and SLAM integration test*

This experiment aims to demonstrate in a real situation, the integration of the people detection node with the SLAM node described in section III-B of this paper. The experiment was conducted in an indoor environment, and the robot's velocity was setup to 10 cm/s and with a frame rate of 15 fps. For this experiment, only the detection node was evaluated.

The figure 15 illustrates the robot running an autonomous mapping while performs a square trajectory through the indoor environment. Two people to be detected stands in different positions on the place. The detected people are labeled in figure as *Person 1* and *Person 2*. The black arrows illustrate the orientation of the robot's path and the green square indicates the start and the end point of the path. Furthermore, it is possible to notice a false positive detection of a chair, indicated by the red square.

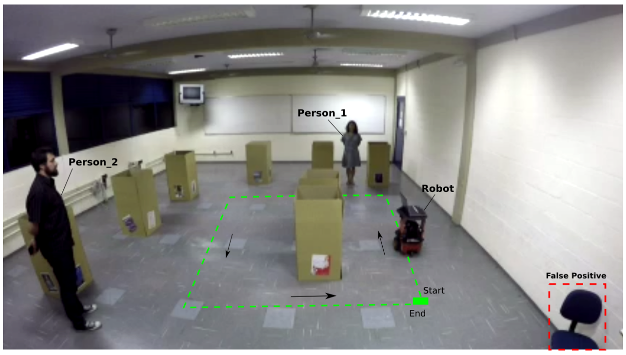


Figure 15. Robot SLAM Path and People Detection Experiment

The figure 15 demonstrates the people detection node's view during the SLAM performing. The top left image in figure 14, shows the bounding box of the detected people, at the top right, it is the RGB image published by the **/camera/rgb/image_rect_mono** ROS topic. In the bottom left of the figure 14 it is an external view, showing the robot following the trajectory, and at the bottom right it is showed the SLAM point computations.

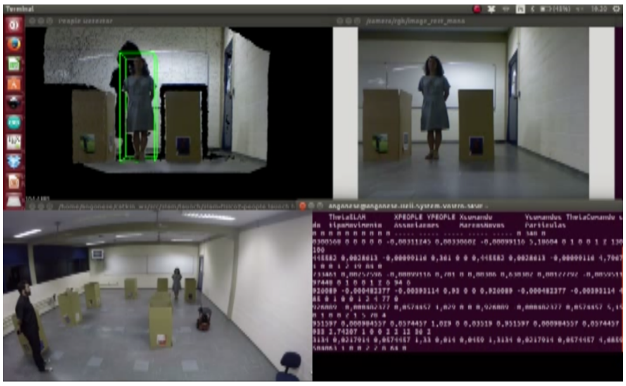

Figure 16. People Detection View of the Experiment: On the top left is the detected people node, at the top right, it is the RGB image. In the bottom left it is an external view and at the bottom right it is the SLAM computations.

The graph of the figure 17, shows the path of the robot and the detected people position related to the robot SLAM trajectory, as illustrated in figure 15. The black arrows indicate the direction of the robot's path and it is possible to note that the graph is 90 degrees rotated to the right. The blue trajectory showed the robot's path if it was defined by odometry. The green trajectory is the correct path defined by the SLAM and followed by the robot. It is possible to perceive the accumulated error of the odometry process, generating an unexpected trajectory. It is also possible to note that the SLAM process is being correctly done, through the green path followed by the robot. The black points are the people's centroids, detected by the robot during the SLAM. The points indicated by the black dotted circles represents the person's centroids detection variation, which occurs due to the robot's movement variation. The red dotted circle is representing the false positive detection of a chair as illustrated in figure 15.

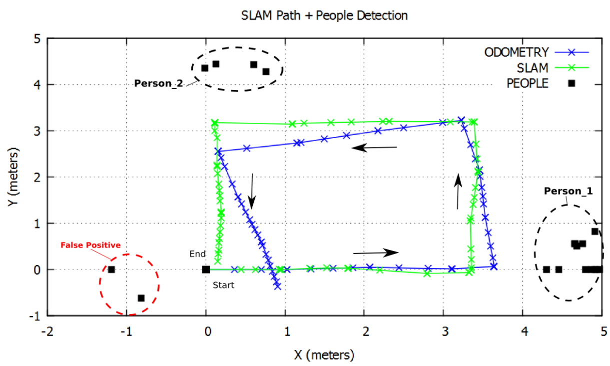


Figure 17. Robot SLAM Path and People Detection Graph

## V. Conclusion and Future Work

This paper presented a people detection and identification system for a robotic platform capable of performing SLAM. The system was implemented in a robot consisting of a 2D laser sensor and an RGBD camera coupled to robotic platform Pioneer 3DX, using the robot operating system ROS. The detection and identification systems were developed as ROS nodes that behave seamlessly with SLAM application node allowing that the robot movement model starts to consider people identification.

The test results of people identification node showed that a Deep Convolutional Neural Network (CNN) has a satisfactory accurateness for people identification in a real time. However, the system is not yet robust to unstructured environments because of frame rate reducing limitation.

As further steps of this research, we propose the implementation of the people detection and identification process as a single ROS node to be integrated with SLAM node. Furthermore, the systems will be implemented in a GPU architecture for better performance, allowing its use in unstructured environments.

## Acknowledgments

This work was supported by *Coordenacão de Aperfeioamento de Pessoal de Nível Superior (**CAPES**)* – Edital *Pro-Estratégia*

## References

[1] F. Silveira Vidal, A. de Oliveira Palmerim Barcelos, and P. Ferreira Rosa, "Slam solution based on particle filter with outliers filtering in dynamic environments," in *Industrial Electronics (ISIE), 2015 IEEE 24th International Symposium on*, June 2015, pp. 644–649.

[2] M. Munaro and E. Menegatti, "Fast rgb-d people tracking for service robots," *Autonomous Robots*, vol. 37, no. 3, pp. 227–242, 2014.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99.

[4] R. Girshick, F. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[5] G. Cheron, I. Laptev, and C. Schmid, "P-cnn: Pose-based cnn features for action recognition," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[6] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5325–5334.

[7] R. Ranjan, V. M. Patel, and R. Chellappa, "A deep pyramid deformable part model for face detection," in *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on*. IEEE, 2015, pp. 1–8.

[8] S. S. Farfade, M. J. Saberian, and L.-J. Li, "Multi-view face detection using deep convolutional neural networks," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 2015, pp. 643–650.

[9] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–4.

[10] A. Vedaldi and K. Lenc, "Matconvnet-convolutional neural networks for matlab," *arXiv preprint arXiv:1412.4564*, 2014.

[11] A. de Melo Neto, P. F. F. Rosa, T. E. A. de Oliveira, and P. C. Pellanda, "Multiple Robots in a Cooperative Task: Exploration and Mapping." in *IROS12 4th International workshop on Planning, Perception and Navigation for Intelligent Vehicles*, Vilamoura, 2012, pp. 79–86.

[12] F. Silveira Vidal, P. F. Ferreira Rosa, A. de Melo Neto, and T. E. Alves de Oliveira, "Cooperating robots for mapping tasks with a multilayer perceptron," in *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*. IEEE, 2013, pp. 4073–4078.

[13] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1627–1645, 2010.

[15] "Kinect tracking precision dataset (ktp dataset)," http://www.dei.unipd.it/~munaro/KTP-dataset.html, accessed: 2016-04-01.

[16] A. T. Angonese and P. F. F. Rosa, "Integration of people detection and simultaneous localization and mapping systems for an autonomous robotic platform," in *Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR), 2016 XIII Latin American*. IEEE, 2016, pp. 251–256.

[17] "Cudaconvnet: High-performance c++/cuda implementation of convolutional neural networks," https://code.google.com/p/cuda-convnet/, accessed: 2016-09-15.

[18] "Theano," http://deeplearning.net/software/theano/, accessed: 2016-08-15.

[19] "Tensorflow: An open-source software library for machine intelligence," https://www.tensorflow.org/, accessed: 2016-09-15.

[20] "Caffe deep learning framework," http://caffe.berkeleyvision.org, accessed: 2016-09-15.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[22] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference*, 2014.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CVPR2015*, 2015.

[25] "Matconvnet pretrained models," http://www.vlfeat.org/matconvnet/pretrained/, accessed: 2017-01-17.