3D object detection with deep learning

Félix Escalona, Ángel Rodríguez, Francisco Gómez-Donoso, Jesus Martínez-Gómez and Miguel Cazorla

Abstract—Finding an appropriate environment representation is a crucial problem in robotics. 3D data has been recently used thanks to the advent of low cost RGB-D cameras. We propose a new way to represent a 3D map based on the information provided by an expert. Namely, the expert is the output of a Convolutional Neural Network trained with deep learning techniques. Relying on such information, we propose the generation of 3D maps using individual semantic labels, which are associated with environment objects or semantic labels. So, for each label we are provided with a partial 3D map whose data belong to the 3D perceptions, namely point clouds, which have an associated probability above a given threshold. The final map is obtained by registering and merging all these partial maps. The use of semantic labels provide us a way to build the map while recognizing objects.

Index Terms—semantic mapping, 3D point cloud, deep learning

I. INTRODUCTION

THE use of appropriate environment representations is needed for most of the current robotic systems. Traditionally, environment representations have been limited to metrical maps that evolved from 2D to 3D with the release of affordable range sensors. In addition to this metric representation, semantic labels can be also used to represent rooms or scene categories. However, the location of relevant elements of the environment should be explicitly provided, which involves human supervision and reduces the adaptability of the system to environment modifications.

In this work, we propose to exploit the representation capabilities provided by available pre-trained deep learning models to automatically label indoor environments. We did not train our own CNN. Instead of that, we adopted the architecture defined by GoogleNet [21] as well as the pre-trained model they provide. This model was trained using the dataset ImageNet 2014 and it obtains a top-5 accuracy of 88.9%. It has to be emphasized that we chose this model over the rest because of its high accuracy rate and the fact that it provides object recognizing features for over 1,000 different classes, which will make our system adaptable and very context independent.

Our approach relies on the use of RGB-D sensors, namely a Microsoft Kinect or Asus Xtion device, suitable for performing a 3D registration of the environment. Over this metric representation, we automatically integrate semantic labels that allow us to determine the most probable location of objects.

This process successfully combines 2D semantic labeling with 3D registration in an automatic fashion. An overall scheme of the proposal can be seen in Fig. 1.

The rest of the paper is organized as follows. In Section II we review some related works and state-of-the-art solutions to the semantic mapping problem and the deep learning techniques. The process for annotating 3D maps based on semantic labels is presented in Section III. Experimental results and the applications of the proposals are discussed in Section IV. Finally, the main conclusions of this work as well as some future research directions are outlined in Section VI.

II. RELATED WORK

Building an appropriate representation of the environment in which an autonomous robot operates is still a widely addressed problem in the robotics research community. This problem is usually known as map building or mapping since maps are considered the most common and appropriate environment representation [22]. A map is useful for robot localization, navigation [5] and path-planning tasks [3], but also for a better understanding of the robot's surroundings [16]. That is, a map may not be limited to metric (e.g. specific poses of objects/obstacles) and topological information (e.g. paths from one place to others), but it can also integrate semantic information (e.g. symbolic representations of objects, expected behaviors for specific locations, or even situated dialogues, to name a few) corresponding to the objects, agents, and places represented on it. In this paper, we propose the use of deep learning techniques to provide semantic information. That information is fused with 3D data in order to obtain a novel map representation, object-oriented.

While the fusion of 3D data and visual information for generating environment representations is not new [8], our proposal presents an important novelty regarding the use of ground truth information. Namely, we rely on expert systems trained from global image annotations instead of pixel level labelling. This increases the range of feasible applications as datasets annotated at pixel level are not easy to generate. Furthermore, the number of classes or categories in the available datasets, such as NYU Depth [19] is notoriously smaller than those with global annotations like ImageNet.

Deep learning architectures have recently revolutionized 2D object class recognition. The most significant example of such success is the CNN architecture, being *AlexNet* [10] the milestone which started that revolution. Krizhevsky *et al.* developed a deep learning model based on the CNN architecture that outperformed by a large margin (15.315 % error rate against the 26.172 % scored by the runner-up not based on deep learning) state-of-the-art methods on the *ImageNet* [18] ILSVRC 2012 challenge.

Félix Escalona, Ángel Rodríguez, Francisco Gómez-Donoso and Miguel Cazorla are with University of Alicante.

E-mail: fescalona@dccia.ua.es, angelillo1992@gmail.com, fgomez@dccia.ua.es, miguel.cazorla@ua.es

Jesus Martínez-Gómez is with University of Castilla La Mancha. E-mail: jesus.martinez@uclm.es

In addition to model generation for solving open problems [4], [15], the release of pre-trained models alongside with the architecture of the networks allows for a direct application of the deep learning systems already generated and tested, as it has been done for the place categorization problem [17]. This is possible thanks to the existence of modular deep learning frameworks such as Caffe [9] that provides easy and fast neural architecture setup and the option of load these pre-trained models. The direct application of pre-trained models avoids the computational requirements for learning them: long learning/training time even using GPU processing, and massive data storage for training data. From the existing deep learning models, we should point out those generated from images categorized with generalist and heterogeneous semantic labels [10], [23]. The use of these models lets any computer vision system annotate input images with a set of semantic labels describing their content, as has been recently shown in [6], [14].

III. USING SEMANTIC LABELING FOR 3D MAPPING

Semantic labeling let images to be described by means of a set of semantic concepts attributed to the scene perceived by the camera. This representation is suitable for human-robot interaction, as semantic terms can be easily included in human-robot interaction processes. The use of semantic labels also facilitates the understanding of robot surrounding, which may help to automatically determine the most appropriate robot behavior in any scenario.

To implement this annotation process we make use of existing deep learning annotation tools. Deep learning techniques, and more specifically Convolutional Neural Networks (CNN [12]), allow the generation of discriminant models while discovering the proper image features in a totally unsupervised way, once the network architecture has been defined. This is possible nowadays thanks to the availability of huge image datasets annotated with large and miscellaneous set of semantic labels, which efficiently permits the training of these discriminative classification models. In this work, we focus on the application of existing CNN models. The definition and building of these CNN models is beyond the scope of this paper, so we refer the reader to [1] for a more detailed view of deep learning in general and, to [9] for a better understanding of these CNN models.

A. Global image annotation using CNN

Let $Label = \{label_1, \dots, label_{|Label|}\}$ be the set of $|Label|$ predefined semantic labels, and I an input image. The direct application of the existing CNN models on I generates a descriptor $d_{CNN}(I) = ([prob(l_1), \dots, prob(l_{|L|})])$, where $prob(l_i)$ denotes the probability of describing the image I using the i -th label in $Label$. This obtains a representation similar to the Bag of Visual Words (BoVW [20], [13]) approach. To train a CNN model, we need to provide both the architecture of the network and the database to be used as training set. The architecture refers to internal details such as the number of convolutional or fully connected layers, or the spatial operations used in the pooling stages. On the other

hand, the training set determines the number of semantic labels used to describe the input image. In this proposal, we take advantage of Caffe [9], a fast, modular and well documented deep learning framework that is widely used by researchers. We opted for this framework because of the large community of contributors providing pre-trained models that are ready to be used in any deployment of the framework.

B. 3D mapping with semantic labels

Given an image, the CNN model provides us with the semantic labels present in the image, so we can expect that the semantic label corresponds to an object present in the image as well. Unfortunately, we are not provided with the position of the image, and therefore the environment, where the object may be placed. So, we propose to move the camera around (left, right, up and down) to look for the limit when the object disappears. Imagine that we move the camera to the left. In that situation, it is expected to have high probability value of a given label (if the object is present in the image). However, when the object disappears, the associated probability decays. If we find the limit when the object appears and disappears, we have a way to determine where the object is (and where it is not). We just need to accumulate all the point clouds where the associated probability to a given semantic label is above a threshold.

We have to implement a mechanism to guide the camera movement. At this moment, this movement is provided by a human but an autonomous system must be developed. The point is that we have to guide the camera using the label probability gradient, searching first for the maximum value of a label and then the limits where the probability label is below the threshold.

This way, when an object is imaged in the current scene the CNN returns a peak probability. As the object disappears of the scene as the camera moves, this probability descends gradually. To deal with this event we propose the application of an hysteresis cycle. We apply the hysteresis cycle as follows: first we set a probability peak threshold Th_1 and a second lower threshold Th_2 . When a probability of a given label exceeds Th_1 , it means that this object is in the scene so this way it enters the hysteresis cycle. From this point we assume the next captures will have this object as well, and we accumulate them to the point cloud of this object. When the probability of this label is below a second much lower threshold Th_2 , it means that this object is no longer in the scene. This event sets the end of the hysteresis cycle so we stop accumulating new point clouds, and we use the last of them to eliminate the exceeding points that do not belong to the object itself but to the background or the rest of the scene. To do so, we calculate the bounding box of this last point cloud that contains information that is no longer related to the given tag. We use this bounding box to remove all the points from the accumulated point cloud that are inside the space described by the bounding box as shown in Fig. 1 leaving, this way, only the points that correspond to the given tag.

As stated earlier, the CNN provides probabilities of finding an object in the scene. That means if we have different objects,

Figure 1. Overall pipeline of the proposal. When a new frame is captured, the RGB-D data is extracted and mapped to a point cloud, then the system registers it to a global coordinate frame. Alongside this process, it extracts the probability of finding objects using a CNN. If the probability exceeds certain threshold and fall into the hysteresis cycle (object is found in the scene), the current point cloud is merged with the point cloud that eventually will hold the 3D object. Otherwise, the bounding box of the current point cloud is calculated and used to crop the integrated point cloud.

which is the common thing, the CNN would provide high probabilities for all of them. As the probability is a value normalized between 0 and 1, if we face this situation, the full probability of finding an object would be distributed over these different objects, causing that none of them reaches the detection threshold. To deal with this problem, we propose the use of a preliminary normalization process.

This mentioned process is as follows. As the CNN provides probabilities for over 1,000 different classes, several of them are not interesting to the environment we are about to label, so we initially select the first N labels which are most frequent. Then, each frame is forwarded to the CNN, and we retrieve the probabilities of these N labels and dismiss the rest. As the probabilities were distributed over the 1,000 classes, we need to redistribute them having in mind only the selected classes. This process is performed in order to remove CNN classifying mistakes involving classes that are not relevant to the current environment and to increase the difference between the probability values to make them more easily distinguishable. Fig. 4 shows the probabilities of 8 labels given a frame before and after the normalization process.

It has to be highlighted that the captured point clouds are registered. This is that we transform every new point cloud from the local camera coordinate frame to a global coordinate frame in order to generate a point cloud that represents an object out of several captured points clouds as the camera moves.

Fig. 1 shows a graphical scheme of the proposed method. The algorithm explaining all the phases of the method is detailed in Algorithm 1. The CNN we are using provides probabilities for over 1,000 different labels, so in order to improve the performance of the system we have also removed all the labels do not providing significant semantic meaning to our

problem. For instance, as we are working in a laboratory, some labels like animals are not appropriated. The use of semantic labels has a clear advantage against classical methods, like those using visual features to identify objects. CNN has the ability of recognizing the same category for two objects even when the visual appearance is complete different.

IV. EXPERIMENTS AND RESULTS

The proposal has been evaluated using a real robot in an academic indoor environment. In order to generate the data sequence for the experimentation, we used a mobile robot fitted with a RGB-D camera, namely an Asus Xtion, on top. The robot performed a 360 degrees rotation over the Z axis taking frames every 5-6 degrees, which provided us with 63 different frames. We assume that the pointcloud registration has been performed (using method like ICP [2], any SLAM technique [7] or other registration techniques such as Rtabmap [11]) so we have all the pointclouds in the same coordinate axis without (or despicable) error. In the experiment, thus we know that the camera rotated 5 degrees over the Y axis between frames (the Y axis corresponds to the vertical one). Therefore, we could easily compute the transformation matrix between consecutive captures. The initial capture was used as the global coordinate frame and we then transformed the following captures applying the convenient rotation matrix to align all the point clouds.

The profiles of the probability values for labels “Desk”, “Desktop Computer”, “File”, and “Microwave” are shown in the Figure III-B. For every label, there is one peak which indicates the position of the object. Fig 3 indicates the resulting pointcloud after removing the rest of the points. Regarding the hysteresis parameters of the algorithm, we settled two

Figure 2. Probability profile of the classes "Desk", "Desktop Computer", "File" and "Microwave" in the first experiment

Figure 3. Labeled objects and their associated point cloud in the first experiment.

thresholds: the peak threshold $Th_1 = 0.4$ and the lower threshold $Th_2 = 0.25$. These thresholds have been empirically selected. Also, we extracted the first $N = 15$ most frequent objects in our test environment and we found that in this subset there are still irrelevant classes, so we manually removed them. Finally, we selected a subset of 8 different objects from the initial set of 15 most frequent ones, as shown in Fig. 4. This selection was made based on the frequency of appearance in the environment of the experiment. For another environment, we must select the most frequent after a first random capture of the robot. Although ImageNet provides more than 1000 categories and we used only 15, those categories will help to work with different environments. For instance, in a kitchen the most frequent labels are different than for a living room.

As seen in Fig. III-B, the profile corresponding to label desktop computer has two peaks. The one reaching 0.3 is below the threshold and then it is not selected. However, a desktop computer is also shown in the pointcloud of the desk (the one in the same position of the previous peak). The reduction of the thresholds would result into a more peaks not corresponding to the actual class of the object. This threshold

Label	Probability	Finally Selected
Refrigerator	219.988	No
Sliding Door	194.273	Yes
Cash Machine	174.483	No
File	162.149	Yes
Dishwasher	116.037	No
Desk	112.222	Yes
Microwave	110.894	Yes
Desktop Computer	78.971	Yes
Monitor	68.187	Yes
Screen	66.352	Yes
Washer	61.099	No
Photocopier	60.067	Yes
Home Theater	49.314	No
Printer	41.118	No
Entertainment Center	30.460	No

Table I
FIRST N LABELS MORE LIKELY TO APPEAR ACROSS THE SEQUENCE, WITH $N = 15$

selection is critical and a further study should be carried out.

It is also worth noted that over the frame 17 of the sequence, the file probability value throws a peak, but as shown in Fig. 3,

Algorithm 1 Algorithm to build the 3D map from semantic labels.

Require: Th_1, Th_2 : thresholds for hysteresis.

```

1:  $\{PC_{label_i} = \emptyset\}$  a set of 3D pointclouds, one for each
   label
2:  $\{hyst_{label_i} = false\}$  a set of boolean variables, one for
   each label
3: loop
4:   Get  $PC_j$ , a 3D point cloud and the associated RGB
   image. The pointcloud must be registered.
5:   Get  $LabelSet = \{(label_k, prob(label_k))\}$ , the set of
   labels and associated probabilities from the RGB image.

6:   for each label  $label_m$  in  $LabelSet$  do
7:     if  $prob(label_m) > Th_1$  then
8:        $hyst_{label_m} = true$ 
9:     end if
10:    if  $prob(label_m) > Th_2$  and  $hyst_{label_m}$  then
11:       $PC_{label_m} += PC_j$  // Add the current pointcloud
      to the map of the given label
12:    else
13:      Get bounding box (BB) of  $PC_j$ 
14:      Remove the points from  $PC_{label_m}$  inside BB
15:       $hyst_{label_m} = false$ 
16:    end if
17:  end for
18: end loop
19: Optional: join all the pointclouds into a single map
    
```

Figure 4. Prenormalization values shows the probabilities of the classes given by the CNN (only the 8 most relevant classes out of the 1,000 full label set) for the frame #6. The postnormalization values shows the probabilities after the normalization process.

Figure 5. Examples of door and file extracted from ImageNet, the dataset used to train the CNN.

that frame corresponds to a door. The appearance of it is very similar to a file, as shown in Fig. 5, so the CNN fails to classify this object successfully. So this is another limitation of the method: if there is an object that it has visual appearance to another one, the CNN could provide a high value to the associated semantic label. Fig. 6 shows the complete map generated after processing four labels (those with the highest probabilities): microwave, desktop computer, desk and file.

Figure 6. Final labeled map obtained from 4 labels.

Figure 7. Probability profile of the classes "Desk", "File", "Microwave" and "Folding chair" in the second experiment

Figure 8. Other examples of final partial maps obtained in the second experiment.

Figure 9. Final labeled map generated by the system for four different objects of the second environment.

A second experiment was carried out. In this case the system run in a different environment. The scenario consists

of a classroom in which we put around several objects such like chairs, a microwave, speakers and others. We run the experiment with the very same parameters established for the first experiment as explained in the first paragraphs of the section IV.

Reviewing the probability profiles of this second experiment, shown in Figure IV, we note that we have a peak in the chair probability at the beginning and in the ending of the data set. This is caused by the spinning motion of the robot: the chair was captured in the first stage as well as in the last stage. Also desks and files are detected all along the dataset, which is correct as the environment is a classroom filled with tables and desks, and files attached to them. We notice that several captures were labeled as microwave. This is an error returned by the CNN subsystem that confuses the monitor and microwave examples.

Some of the results thrown by the system could be seen in the Figure 8. Several objects such as desks, files, chairs and a microwave were detected and segmented properly as shown in Figures 8(a), 8(b), 8(c), and 8(d). A final map was generated by the system in which put together all the single object point

clouds in a final labeled map, as shown in Figure 9.

V. FUTURE WORK

Improving the method is twofold: including 6 degrees of freedom and isolating the target objects with more precision. To achieve this goal, several new problems must be solved.

Firstly, we do not have precise articulations that tell us the exact transformation between two consecutive point clouds, so we must use a registration method for the purpose of fitting the points and obtaining a good 3D scene with the correct location of the objects.

Secondly, the images taken from different points of view of the same object could be misclassified by the convolutional neural network, so we may establish some kind of filter to avoid several situations in which the object is not detected suddenly.

Figure 10. First results of the 6-dof algorithm.

In Figure 10 we can see the first results of the new approach including 6 degree of freedom, after making several cuts above, under and left side of the microwave from one point of view.

VI. CONCLUSION

In this paper we have presented the first approach to a new perspective for semantic 3D map building. Instead of recognizing objects, we leave this task to an oracle which provides us with the objects present in the image and the probability of those object to be present. We have used a convolutional neural network as an oracle, using Caffe and the model from Imagenet.

With that information, we process the 3D data, incorporating new pointclouds if the object probability is above a given threshold (applying hysteresis). If the probability is below that threshold, the bounding box from the current pointcloud is used to remove the points inside the bounding box from the accumulated point cloud. Results are convincing but more development must be done to get a full valid system. We have showed results using only a rotation (even a translation should provide similar results) around Z axis. In order to get a full 6 degrees of freedom system, several adjustments and experiments must be done.

ACKNOWLEDGMENT

This work was supported by grants DPI2013-40534-R, and TIN2015-66972-C5-2-R of the Ministerio de Economía y Competitividad of the Spanish Government, supported with Feder funds, and by Consejería de Educación, Cultura y Deportes of the JCCM regional government through project PPII-2014-015-P.

REFERENCES

- [1] Y. Bengio. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.
- [2] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [3] P. Bhattacharya and M.L. Gavrilova. Roadmap-based path planning - using the voronoi diagram for a clearance-based shortest path. *IEEE Robot. Automat. Mag.*, 15(2):58–66, 2008.
- [4] L. Bo, X. Ren, and D. Fox. Unsupervised feature learning for rgb-d based object recognition. In *Experimental Robotics*, pages 387–402. Springer, 2013.
- [5] O. Booi, B. Terwijn, Z. Zivkovic, and B. Kröse. Navigation using an appearance based topological map. In *International Conference on Robotics and Automation*, pages 3927–3932. IEEE, 2007.
- [6] G. Carneiro, J. Nascimento, and A.P. Bradley. Unregistered multiview mammogram analysis with pre-trained deep learning models. In *Medical Image Computing and Computer-Assisted Intervention*, pages 652–660. Springer, 2015.
- [7] M. Cazorla, P. Gil, S. Puente, J. L. Muñoz, and D. Pastor. An improvement of a slam rgb-d method with movement prediction derived from a study of visual features. *Advanced robotics*, 28(18):1231–1242, 2014.
- [8] A. Hermans, G. Floros, and B. Leibe. Dense 3d semantic mapping of indoor scenes from rgb-d images. In *International Conference on Robotics and Automation*, pages 2631–2638. IEEE, 2014.
- [9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 675–678, New York, NY, USA, 2014. ACM.
- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [11] M. Labbé and F. Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In *International Conference on Intelligent Robots and Systems*, pages 2661–2666. IEEE, 2014.
- [12] H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM, 2009.
- [13] J. Martínez-Gómez, V. Morell, M. Cazorla, and I. García-Varea. Semantic localization in the PCL library. *Robotics and Autonomous Systems*, 75, Part B:641 – 648, 2016.
- [14] V.N. Murthy, S. Maji, and R. Manmatha. Automatic image annotation using deep learning representations. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 603–606. ACM, 2015.
- [15] N. Neverova, C. Wolf, G.W. Taylor, and F. Nebout. Multi-scale deep learning for gesture detection and localization. In *Computer Vision-ECCV 2014 Workshops*, pages 474–490. Springer, 2014.
- [16] A. Pronobis, O. Martinez Mozos, B. Caputo, and P. Jensfelt. Multi-modal semantic place classification. *The International Journal of Robotics Research*, 2009.
- [17] J.C. Rangel, M. Cazorla, I. García-Varea, J. Martínez-Gómez, É. Fromont, and M. Sebban. Scene classification based on semantic labeling. *Advanced Robotics*, pages 1–12, 2016.
- [18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, An. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

- [19] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012*, pages 746–760. Springer, 2012.
- [20] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003.
- [21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. 2015.
- [22] S. Thrun et al. Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, pages 1–35, 2002.
- [23] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495, 2014.