The Visual Computer
Semantic scene synthesis: Application to assistive systems
--Manuscript Draft--

Manuscript Number:	TVCJ-D-20-00852R1
Full Title:	Semantic scene synthesis: Application to assistive systems
Article Type:	Original article
Keywords:	Semantic labeling; Depth image; Assistive systems; Point cloud classification; Deep learning; Scene synthesis
Corresponding Author:	SLIMANE LARABI, Professor usthb university Algiers, El ALIA ALGERIA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	usthb university
Corresponding Author's Secondary Institution:	
First Author:	Chayma ZATOUT
First Author Secondary Information:	
Order of Authors:	Chayma ZATOUT
	SLIMANE LARABI, Professor
Order of Authors Secondary Information:	
Funding Information:	
Abstract:	The aim of this work is to provide a semantic scene synthesis from a single depth image. This is used in assistive aid systems for visually impaired and blind people that allows them to understand their surroundings by the touch sense. The fact that blind people use touch to recognize objects and rely on listening to replace sight, motivated us to propose this work. First, the acquired depth image is segmented and each segment is classified in the context of assistive systems using a deep learning network. Second, inspired by the Braille system and the Japanese writing system Kanji, the obtained classes are coded with semantic labels. The scene is then synthesized using these labels and the extracted geometric features. Our system is able to predict more than 17 classes only by understanding the provided illustrative labels. For the remaining objects, their geometric features are transmitted. The labels and the geometric features are mapped on a synthesis area to be sensed by the touch sense. Experiments are conducted on noisy and incomplete data including acquired depth images of indoor scenes and public datasets. The obtained results are reported and discussed.

The Visual Computer

Title: Semantic scene synthesis: Application to assistive systems.

Authors: Chayma Zatout, Slimane Larabi.

We thank the Chief Editor and Reviewers for reviewing our work and giving useful comments.

Before responding to the comments, we would like to notify the Reviewers that, as you suggested, we investigated the use of another type of deep learning models (named PointNet). The latter reached an accuracy of up to 96% while the former reached only 72%. As a result, the section 4.1 "Semantic classification" and the section 6.3 "Semantic classification" have been modified accordingly.

Our responses to the reviews can be found one by one below:

Reviewer #1

General comment: The authors have investigated is a very interesting idea for assistive systems for the visually impaired. This work is mostly application-based. The authors propose an application framework for creating scene understanding using deep neural networks. The scene understanding is done in two stages: the classification of various objects captured by the depth image followed by semantic labeling of the data similar to the Braille and Kanji systems.

Comment 1: The idea for proposed semantic labeling is neat. But have authors considered testing it for a real scenario? How will this system be put in use? How difficult is it for the person comfortable with Braille/Kanji system to adapt to a new proposed semantic labeling?

Response 1: Yes, we are considering testing it for real scenarios. In fact, we are willing to implement the presented device and then we will carry out extensive clinical studies.

Although the labels are inspired by Braille and Kanji systems, understanding and memorizing these labels do not require the user to be comfortable with them. The user is only requested to memorize the proposed labels and recognize them by touching the tangible area.

Comment 2: Why is a depth camera required for this task? Why not just go with RGB-semantic segmentation using deep learning? Won't it make the system more accessible?

Response 2: The system can be adapted to use an RGB camera as the input sensor; however, one more step that consists of estimating the depth image will be required. In the visually impaired assistive systems, knowing the distance away from objects and the geometry of the captured scene are important.

Comment 3: Figures are not presented properly. There are no labels for axes in Fig. 10, 11, 12, 13, 14, 15. Results in figure 14, should be placed side by side. In the current draft, these appear one above the other.

Response 3: The Figures have been updated with the labels and the results have been placed side by side.

Comment 4: It looks like in Figure 17, the system predicted that it is looking at a chair, but the segmentation algorithm has only segmented a table in the scene. Was this an error or did the system predicted it incorrectly? This should be specified clearly.

Response 4: We agree with the comment, we explained as follows:

"The system has predicted incorrectly the class of the table that was poorly segmented (we only used background removal to simulate the cropped data)."

Comment 5: Why is figure (2) book marked? Was it taken from somewhere? The citation for the source of the figure is missing. It should be provided in the caption or as a footnote on the same page.

Response 5: We are sorry for the figure. The figure is updated.

Comment 6: *The authors should consider making the data and code openly available on platforms such as GitHub.*

Response 6: Our data and code are now in the Github repository as suggested. Here is the link:

https://github.com/ChaymaZatout/Semantic-scene-synthesis

Comment 7: *In section 5.3, the authors talk about various network architecture and training. In 5.3.3, it mentions "some" layers have been removed from VGG-16? Which are these layers is not specified. Section 5.3 does not explain how is the network trained or which optimizer is used for training the network.*

Response 7: We have taken in consideration this comment for this new version. We mentioned the input size; as for the training setting, we mentioned that is the same with the original paper.

Comment 8: *In its current form, the paper is very lengthy. The number of pages is 26. Please consider shortening it. This paper requires a major revision.*

Response 8: We revised the paper and we shortened it as suggested by the reviewer (from 26 pages to 22 pages).

Reviewer #2

General comment: An entire pipeline is proposed from the depth image to the touch-based user feedback. All individual steps are reasonable. Selecting base classes based on their use (sitting, putting something in, etc.) and combining these with size information to reach a total of 15 classes are both good decisions that improve the semantic meaning of the classification. The encoding system for relying the object detection information to the user is well through-out.

Comment 1: *No comparison takes place with another method for the object classification module. Even though the problem is slightly different, the results could have been compared to methods that use RGB images. Significant space is taken up describing the encoding system that a user would receive, yet there is no evaluation of this system.*

Response 1: The results are now compared to the presented works in the related work section.

Comment 2: *Plot axis should be labeled. Also, there are too many training trajectory plots.*

Response 2: The Figures have been updated with the labels. We have taken this in consideration; the new results have at most 2 trajectory plots.

Comment 3: *Some information is unnecessary considering this is a Computer Visual journal and takes up too much space. For instance, the distinction between "online mode" and "offline mode" for the classification module, the scaling as a preparation step for the CNN, the general SOTA results of deep learning architectures (in the beginning of section 3.2), the statement that "we took a pretrained model using data from the ImageNet challenge and then, we investigated the fine-tuning technique before applying transfer learning" (saying a pretrained model is fine-tuned is enough).*

Response 3: We would like to thank you for this comment. We have taken it in consideration for the whole paper.

Comment 4: *The abstract does not explain or summarize the paper. I would consider writing a longer abstract where the motivation, contributions and pipeline components are each touched upon. It is also not clear from the abstract that the system is meant to aid visually impaired users.*

Response 4: The abstract has been updated as you suggested.

Comment 5: *It is stated that an i5 processor is used. If this is to be included, then the entire name should be used.*

Response 5: After that, we executed our proposed system on a laptop having Intel Core i5-7200U CPU (2.50GHz × 4) and 4GB as processor and RAM respectively.

Comment 6: *This sentence is unclear: "The DBSCAN is robust to noise, during clustering the noise is detected and classified as noise (fig. 9 third bottom image)."*

Response 6: The sentence has been reformulated: Furthermore, the DBSCAN is robust to noise: during the clustering, the outliers are detected and neglected and thus, the clustering process is not affected by them.

Comment 7: *The notation for figures is not consistent. "Figure", "Fig." and "fig." are all used.*

Response 7: We opted for "Figure" for the entire paper.

Comment 8: *It is not stated which layers are removed from the VGG-16 architecture.*

Response 8: In the revised paper we considered a new model architecture, we did not remove any layers.

Comment 9: *In Fig. 14, the plots are on the same column, yet the caption references the "right" and "left" plots.*

Response 9: The plots have different colors and the results have been placed side by side.

Comment 10: *Some orthographic or grammatical errors.*

Response 10: The paper has been thoroughly proof read and revised. A sincere effort has been made to handle all expression problems and grammatical errors.

Reviewer #3

General comment: *Thank the authors for the submission of the manuscript entitled "Semantic scene synthesis: Application to assistive systems". Presentation of the research work is precise and work is well defined. The 'abstract' and 'introduction' is well written. The state-of-the-art is discussed in the article and well connected with the overall situation and context.*

Comment 1: *It would be much better if the authors give separate 'motivation' and 'objective' for the work. That would give more understandability of the model.*

Response 1: We have separated the 'motivation' and 'objective' for the work (section 2.3).

Comment 2: *The authors have used DCGD algorithm, some description should be needed about the DCGD algorithm.*

Response 2: We have added a subsection (section 3.1) for the DCGD algorithm.

Comment 3: *The authors have used data from RMRC challenge and GDIS dataset. Some snapshot or instances of the datasets could be given for better understanding.*

Response 3: We have added some instances for both dataset (Figure 10).

Comment 4: *More public datasets and deep learning techniques could be given to test the performance of this model.*

Response 4: We have taken this in consideration, we replaced the CNN-based model with PointNet, a point cloud based model.

Comment 5: *For segmentation the authors have used K-means, mean-shift, fuzzy-clustering and DBSCAN clustering algorithms. A short description of these algorithms can be given for better understanding.*

Response 5: We have added a short description for each algorithm:

"K-means, mean-shift and fuzzy-clustering are centroid based algorithms: the first divides the input samples into K separate groups with equal variance while minimizing a given criterion. The mean-shift algorithm separates the input samples into blobs with smooth density. As for the fuzzy-clustering which is based on fuzzy-logic, it assigns for each sample a probability corresponding to each cluster. More the probability is high the more the sample is near to the cluster. One of the disadvantages of these algorithms is the fact that they require the number of clusters as input which is unknown when dealing with real-world data. On the other hand, the DBSCAN algorithm seeks to separate the samples into high density clusters with low density areas. In addition, it is a non-parametric algorithm, the number of clusters does not have to be defined beforehand. Furthermore, the DBSCAN is robust to noise: during the clustering, the outliers are detected and neglected and thus, the clustering process is not affected by them."

Comment 6: *Authors don't show the experimental setting, such as batch size, epoch, learning rate, training time and running environment in details and precise manner.*

Response 6: We have taken in consideration this comment for this new version. We mentioned the input size; as for the training setting, we mentioned that is the same with the original paper.

Comment 7: *The authors could have given a separate section named 'Future Scope' for this work.*

Response 7: We have added the 'Future Scope' section (section 7).

Semantic scene synthesis: Application to assistive systems

Chayma Zatout · Slimane Larabi

Abstract The aim of this work is to provide a semantic scene synthesis from a single depth image. This is used in assistive aid systems for visually impaired and blind people that allows them to understand their surroundings by the touch sense. The fact that blind people use touch to recognize objects and rely on listening to replace sight, motivated us to propose this work.

First, the acquired depth image is segmented and each segment is classified in the context of assistive systems using a deep learning network.

Second, inspired by the Braille system and the Japanese writing system Kanji, the obtained classes are coded with semantic labels. The scene is then synthesized using these labels and the extracted geometric features.

Our system is able to predict more than 17 classes only by understanding the provided illustrative labels. For the remaining objects, their geometric features are transmitted. The labels and the geometric features are mapped on a synthesis area to be sensed by the touch sense.

Experiments are conducted on noisy and incomplete data including acquired depth images of indoor scenes and public datasets. The obtained results are reported and discussed.

C. Zatout
RIIMA Laboratory, USTHB University, 16111 Algiers, Algeria
Tel.: +213-21-247607
Fax: +213-21-247607
E-mail: slarabi@usthb.dz

S. Larabi
The same university.

1 Introduction

In order to accomplish daily tasks, people involve their five senses, namely sight, hearing, taste, smell and touch. Being deprived of one of these senses will complicate the process of a given task; it will reduce human autonomy, independence and even privacy; the visually impaired find difficulties in their daily life.

With the limitations of the classical aid systems such as white canes, guide dogs and personal assistants; and with the evolution of technology, many commercial and noncommercial aid systems were proposed in the last decades. Generally, these latter rely on image processing, artificial intelligence techniques and external sensors in order to offer help for the visually impaired and blind people to improve their independence in many applications.

To transmit instructions, scene description or any other generated output, most of the assistive systems use audio-based or vibration-based output devices. It turns out that these latter hold hearing and are not too informative. Hence, the necessity of providing a semantic labeling for scene understanding that can be exploited by the touch sense.

In this work, we propose a framework for semantic scene synthesis. From the depth image, the 3D scene is down-scaled and semantically mapped into a synthesis area using the computed labels and the extracted geometric features of the input point cloud. Two main modules are proposed: the classification module and the semantic labeling module. The first module is based on a deep learning architecture to classify depth image segments into seven semantic classes. The semantic labeling is inspired from Braille and Kanji systems. This latter is mapped into a touch-based synthesis area that can be used for many applications such as in assistive systems for visually impaired and blind people.

The remaining sections are structured as follows: in section 2, we present related works to objects classification, scenes understanding and semantic labeling for assistive systems. An overview of the proposed system is presented in section 3. In sections 4 and 5, we describe our approaches for the classification and the semantic labeling module respectively. Conducted experiments are reported and discussed in section 6. Finally, we conclude with the future works (section 7) and a conclusion (section 8).

2 Related Works

Visually impaired aid systems consist of a set of techniques whose goal is to enhance the visually impaired life in different activities. These systems can be traditional like white canes, guide dogs or personal assistants; sophisticated by involving advanced technologies and computer science techniques [1] or hybrid [2][3][4]. The sophisticated systems process the received data from the real-world using sensors and transform it into instructions and signs that can be understood by the visually impaired people. They use depth or RGB sensors,

image processing techniques, computer vision and machine learning. In this work, we focus on scene understanding and semantic labeling.

2.1 Scene understanding and object recognition

Object recognition in complex scenes is still a challenging problem. In the recent work [39], they recognized objects based on alignment of convex hulls of the detected segments in a depth image. Each computed convex hull was then compared with convex hulls of target 3D object models or their parts. This alignment is performed using the Convex Template Instance descriptor. In visually impaired assistive systems, detecting objects is important in most applications. Knowing their nature will provide the ability to deduct additional information such as scene understanding, auto positioning and creating free space by moving some kinds of objects like chairs. In [5], using a depth camera, they proposed a wearable system based on a linear classifier to classify the point cloud features into: chairs, tables, stair-up, stair-down and walls. The obtained class is coded and mapped into a braille display. In [6], they proposed a system that uses Visual Odometry, Region-growing and Euclidean cluster extraction, and depth data to determine if the horizontal planes are a valid step of a staircase. In [7], they implemented scene understanding using deep learning techniques. The scene is captured using an RGB-D camera and the results are displayed using an earphone and a smartphone that serves as a haptic device. They adapted FuseNet [8] and GoogleLeNe [9] to provide semantic segmentation and orientation instructions respectively. The semantic segmentation model was designed to predict 40 different classes; however, these latter are transmitted using an audio device. The authors in [10], segmented the input point cloud based on the cascaded decision tree using RGB-D images. The system only classifies a given segment, whether it is the ground, a wall or a table (a horizontal plane that is not the ground). As in [11], they proposed a lightweight CNN architecture that is able to be executed on smartphones for traveling assistive systems. They adopted PeleeNet for object detection to cover 80 different classes using the RGB images as input.

2.2 Semantic labeling

In [12], they proposed a navigation system based on RGB image processing. The system transmits the generated scene and the navigation instructions on the Senseg TM device. They used the electrostatic signs to generate codes and to form textural instructions for the visually impaired and blind people. They also used colors to encode additional visual feedback for the sighted and with low vision people.

Authors in [5], encoded the considered object classes using the first character of the class's name: o, c, t and the space character to represent obstacles, chairs, tables and free spaces respectively. They used a braille device to transmit the occupancy grid to the user and to deliver the object's label in the

Braille system. These codes are simple to understand, but it can be ambiguous while covering a large set of objects, especially with objects having the same first character.

In our previous work [13], we proposed a semantic labeling for scene understanding since after understanding the scenes' components, a human-being can accomplish other tasks such as navigating. We mapped the detected planes into cylinders having a specific height and radius. These latter are mapped into a trapezoid area at a specific location. By touching this area, the user can locate objects from the free space that allows him to have an idea about their characteristics, namely their heights and their areas. The proposed semantic labeling does not reveal the object's nature, it only gives an impression about the given object.

2.3 Motivations and Objectives

The state of the art systems present many limitations:
- In indoor navigation, for example, they provide only a global description of the captured scenes (free space and obstacles without describing their nature). In some proposed systems when objects are detected, they usually use an RGB camera as an additional input sensor [14]. In other works, they provide obstacle classification, but by considering only a few classes such as classifying the scene components into the floor, objects that are parallel to the floor and objects that are perpendicular to floor without considering other features such as the object's height and its occupied area.
- The system's output is usually transmitted by an audio device. This latter suffers from some limitations as explained in [13] and in [15] works.
- The proposed semantic labelings are less informative and can generate ambiguity and do not consider some geometric features that can be helpful [5].
- The actual aid systems are vibration-based or audio-based. For the visually impaired, the hearing replaces the sight in different tasks such as detecting some events, thus, the importance of releasing it.
- The visually impaired use the touch for learning geometries, recognizing objects and human faces; and even for reading.

As a consequence, our objective through this work, is to propose an end-to-end assistive system that uses only the depth image as an input and it generates labels on a tangible area. This system provides a scene description designed specially for the visually impaired: it allows them to locate objects and to recognize the geometry of the captured scene. The tangible area is used as an output device to replace the traditional output device and thus the hearing is released and the labeling is more informative.

3 System overview

Our proposed system (Figure 1) takes a single depth frame captured by a head-mounted depth camera as an input. After detecting the ground using
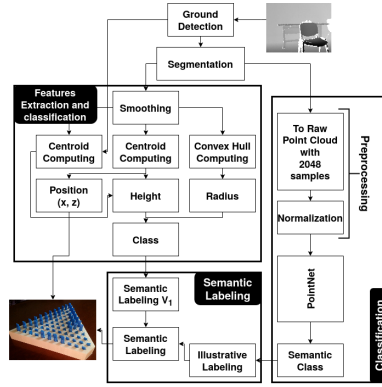
Fig. 1 The three essential modules of our proposed framework: Features extraction module, Classification module and the module that generates semantic labeling.

the DCGD (Depth-Cut based Ground Detection) algorithm [13], the occupied space is extracted and segmented. Each computed segment is then fed to the deep network to perform object classification. The feature extraction module computes geometric features for each segment such as the object's height. After that, the provided class and features are used to generate semantic labels. Finally, the captured scene is synthesized based on the features and labels computed previously. The system can be adapted to use an RGB camera as the input sensor; however, one more step that consists of estimating the depth image will be required. In the visually impaired assistive systems, knowing the distance away from objects and the geometry of the captured scene are important.

3.1 Ground detection

The DCGD algorithm [13] consists of computing a set of depth cuts from a given depth image. For a given depth z_i, the selection on each column of the depth image pixels having $z = z_i$ will define the images of 3D points at distance z_i. These 3D points define a parallel plane (Π_i) to image plane cutting the 3D scene (Figure 2). The selected pixels on each column having the minimal value of the $y - component$ define the image of intersection (G_i) of the plane (Π_i) with the 3D scene. From the initial distance z_0 to the final distance (maximal authorized) z_f, for each step δz, the curve $G_i, i = 0, n$ is computed at $z = z_0 + i \times \delta z$, such that $z_f = z_0 + n \times \delta z$. The second step consists of removing pixels from (G_i) that corresponds to objects. Each cut is divided into sub-cuts labeled either "concave" or "convex". The "convex" parts represent objects on the floor. The ground is then detected by iteratively removing the convex of the current cut to keep only concave parts which will, at the end of the process, represent the ground.
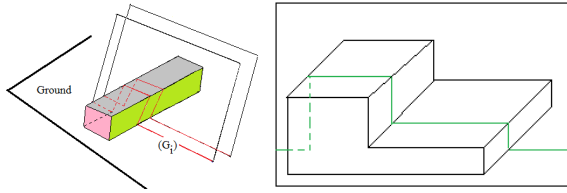
Fig. 2 (Left)The 3D points at a given depth with minimal $y-$coordinate (case where $xz-$plane is parallel to the ground) (colored in green).(Right) The obtained curve (G_i) for two cutting planes.

3.2 Point cloud segmentation

In this work, we are interested in detecting and representing coarse information, i.e. large segments that represent salient objects. Detecting and transmitting coarse information in visually impaired assistive systems is important, knowing the scene architecture and how objects are arranged can be useful to accomplish many daily tasks such as scene understanding and navigation. Accordingly, in order to segment the obtained point cloud into coarse segments, clustering algorithms can be used for irregular object segmentation as mentioned in [27]. K-means [28] [29], mean-shift [30], fuzzy-clustering [31] and DBSCAN [32][33][34] clustering algorithms are widely used for point cloud segmentation.

K-means, mean-shift and fuzzy-clustering are centroid based algorithms: the first divides the input samples into K separate groups with equal variance while minimizing a given criterion. The mean-shift algorithm separates the input samples into blobs with smooth density. As for the fuzzy-clustering which is based on fuzzy-logic, it assigns for each sample a probability corresponding to each cluster. More the probability is high the more the sample is near to the cluster. One of the disadvantages of these algorithms is the fact that they require the number of clusters as input which is unknown when dealing with real-world data. On the other hand, the DBSCAN algorithm seeks to separate the samples into high density clusters with low density areas. In addition, it is a non-parametric algorithm, the number of clusters does not have to be defined beforehand. Furthermore, the DBSCAN is robust to noise: during the clustering, the outliers are detected and neglected and thus, the clustering process is not affected by them.

In case of point cloud coarse segmentation, the DBSCAN is more suitable as shown by Figure 3. To reduce the temporal complexity, we applied a downsampling on the input point cloud. A pass-through filter can be applied for noise removal; its parameters are fixed according to the characteristics of the depth camera. In our case, we only considered points with a depth between $800mm$ and $4000mm$ apart. After the segmentation step, each segment is injected into the classification module and the feature extraction and classification module.
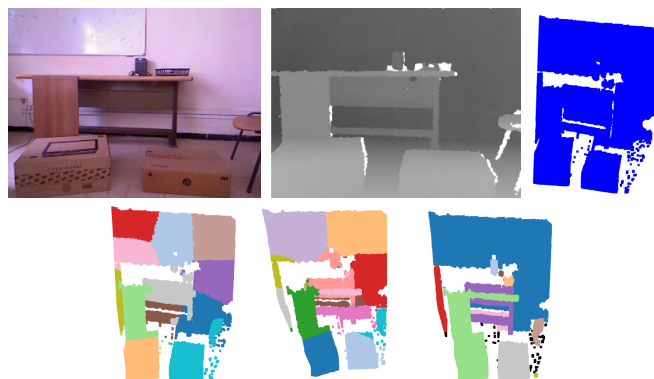
Fig. 3 From left to right:(Top) color image, depth image and occupied space's point cloud. (Bottom): K-means, Mean-shift and DBSCAN (the black points represent the detected noise).

In the next two sections, we will present in detail the main system components, namely the point cloud classification module, that is responsible for computing the geometric features and finding the point cloud nature using a deep neural network. Then, the semantic labeling module that is responsible for designing and synthesizing the scene using the objects classes that may serve as an output interface for assistive systems.

4 Point cloud Classification in the context of assistive systems

Our aim is to provide object classification in the context of assistive systems. In order to cover a significant number of classes in a simple way, we consider salient objects (large and medium objects) and regroup them into 7 semantic classes that include 16 object classes, namely chairs, beds, sofas, benches, stools, tables, desks, night-stands, dressers, wardrobes, shelves, bathtubs, toilets, windows, doors and stairs. These classes do not only contain semantic meaning, but the objects belonging to the same class have also a similar geometric shape while preserving some characteristics. The first class represents objects that we sit on, it includes chairs, stools, beds, sofas and benches. The second class represents objects that we put something on, it includes tables, night-stands and desks. The third class represents objects that we put, hide or arrange something in, it includes dressers, wardrobes and shelves. As for the fourth, the fifth and sixth classes, they represent bathtubs and toilets; windows and doors respectively. The last class represents stairs and since the stairs can be dangerous for the visually impaired and blind people, making the difference between the stairs leading to upstairs and stairs leading to downstairs is important.

In addition, this module includes 3 different sub-modules. The first sub-module computes the point cloud semantic class based on classifying its global

features computed using a deep neural network. The second sub-module computes the point cloud geometric features. As for the last sub-module, it combines the output of these sub-modules in order to provide the final classes and provide them for the semantic labeling module.

4.1 Semantic classification

We call the semantic classification/classes, the classification/classes provided by the deep neural network in order to distinguish between this later and the final classification that is mapped on the scene synthesis.

Deep learning architectures showed considerable improvement in many fields, especially with RGB data after introducing the Convolutional Neural Networks, CNNs. Current works investigate the 3D object classification using volumetric data [18][16][19] or point cloud [20][21][22][23] as the input. The advantage of using point clouds based networks instead of CNNs based or volumetric based networks is the fact that point clouds are irregular, most of the time sparse and permutable.

Note that, in this work, we are not designing a novel architecture. However, using our approach of merging the objects classes that are geometrically similar and have the same function increased the accuracy with 5% on the test set. The obtained results are reported, compared and discussed in the experiments section. To classify a given point cloud, we trained the PointNet network [23], a state of the art architecture. PointNet is a deep multi-layer perceptron network that is designed for 3D raw data. It is invariant to permutation and transformation due to the use of max pooling, a symmetric activation function, and T-net that ensures pose normalization. It is also robust to small noise and incomplete data (with a small portion), but it is not able to capture fine local patterns. This limitation will not highly affect our system since we only consider salient objects (for the time being). Although several architectures have been proposed over time such as [20] [21], PointNet is still used in different systems that involve point clouds [44] [43].

4.2 Geometric features extraction

In addition to the discussed object classification, we extract from the computed convex hull geometric features, namely object's height and its occupied area as explained in [13]. This is done in the online mode, the offline mode is not needed for this classification. After the segmentation process, extracting geometric features and the classification is computed in parallel with the object classification. This object description is useful for the visually impaired and blind people, it helps them to have an impression about their surroundings.

4.3 Point cloud classification

We combined these two classification methods, i.e. the classification using deep learning and the classification using the geometric features to provide rich information. For each object, we provide its semantic class, its height class and its occupied area class. When the deep learning model predicts the object class with low probability, only the second classification is maintained. In this way, we are sure to provide an accurate description even if the deep learning model fails to predict the right semantic class.

In addition, combining these classes allows the distinction between objects belonging to the same semantic class. Chairs, beds, sofas and benches are objects that we sit on; however, they are different in their forms and occupied area: beds are generally a large object (3rd class regarding the occupied area), chairs have small surface (1st class regarding the occupied area) and benches and sofas are medium objects (2nd class regarding the occupied area). Dressers, night-stands and shelves are all objects that we arrange something in; however, they have different heights and occupied areas: dressers are represented by the 3rd class regarding the height, night-stands are represented by the 2nd class, but shelves are usually medium objects and night-stands are usually small.

5 Object semantic labeling and scene synthesis

In this section, we first introduce our proposed semantic labeling that represents the class of each detected object. Then, we describe how the perceived 3D scene is synthesized into a specific area while taking into account the located ground, the semantic label and extracted features of each 3D object.

5.1 Semantic labeling

In our previous work [13], we proposed our first semantic labeling: each horizontal plane in the scene was represented as cylinders with an associated height and radius that allow representing the occupied area and the height of the associated object respectively. This allows the user to have an impression about his surroundings: free space, small objects, medium objects, large objects and their heights, but it does not provide the object's shape nor its nature: if it is a chair, a table, etc.

In this work, we propose an improved semantic labeling which takes into consideration the nature of the objects. It is inspired by the **Braille system** and **Kanji** (the Japanese writing system)(Figure 4). An alphabet in the Braille system is represented by a cell that is provided with raised dots. Each cell contains at most six raised dots. **Kanji** is a Japanese writing system that is inspired by logographic Chinese characters. Some Kanji letters that represent some objects are driven from nature, i.e. these objects' shape in real-world; it's the case for 'mountain' as shown in Figure 4 (right) first line.
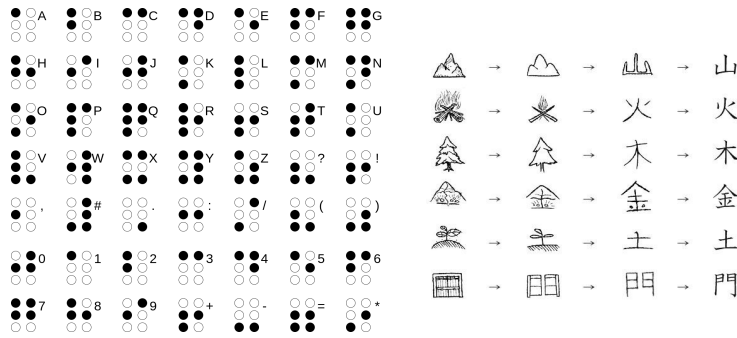
Fig. 4 (Left) The Braille system. (Right) Kanji, the Japanese writing system.
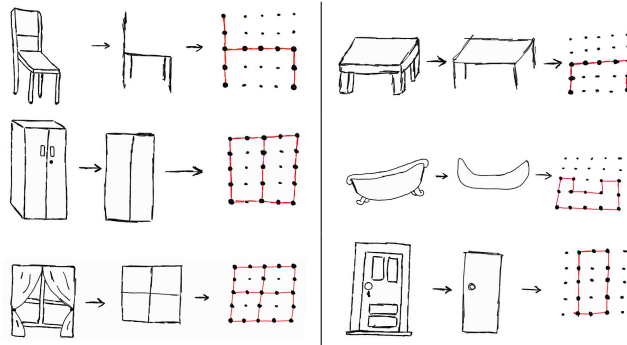

Fig. 5 Our proposed semantic labeling from the first class to the last class, respectively (From top to down). We first drew the selected objects in the real-world (Left) and then, we derived shapes recursively until we obtained the current semantic labeling (From left to right).

In order to draw our illustrative semantic labeling, we designed cells with at most 25 raised dots. The cell's shape can be revised depending on the precision of the synthesis area: if the synthesis area is rich in pins, we can design cells with more raised dots. We suggest to use cells such that the shape can be touched only by a single finger to avoid ambiguity.

On the other hand, the illustrative labeling is derived from the object's shape in the real-world. For each semantic class, we chose the object that is the most close to the class' meaning: we chose a chair, a table, a dresser, a bathtub, a window and a door to represent the proposed classes in section 4 respectively as shown in Figure 5. To ensure the user's safety, we provide two different labels for the stairs leading to upstairs and stairs leading to downstairs as shown in Figure 6. In this way, by touch, the user will understand if he is about to climb the stairs or about to go downstairs.

The described labeling is used alongside with the first labeling to enrich the scene description. The reason we combined the two labels instead of creating an illustrative label for each object, is to reduce the number of possible illustrative labels since it can be deducted from its height and its occupied
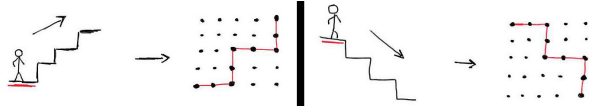
Fig. 6 The proposed semantic labeling for stairs. Stairs leading to upstairs and leading to downstairs, their chosen labels, their positioning on the synthesis area.

area as explained in section 4.3. Note that although the labels are inspired by Braille and Kanji systems, understanding and memorizing these labels do not require the user to be comfortable with them.

5.2 Scene synthesis

After detecting the ground and classifying the computed segments, the system provides the ground pixels and a set of segments. Each segment is associated with its label and its extracted geometric features. These latter will be used for scene synthesis.

The area of the scene synthesis has a trapezoid shape that corresponds to the scaled view field of the camera. This area is covered by a dense number of pins that represent the raised dots in the Braille system. The pin is set up to **5** different levels: The level zero is used to represent the holes on the ground. The level 1 represents the ground or the neutral element in some applications. The height of the illustrative labels that are mapped in the center of the convex hull, is set to the computed level (i.e. level 2, 3 or level 4) and the height of the remaining area of the convex hulls associated with the segment is set to level 2.

The parameters of the depth camera are used to map the view field into an area of scene synthesis (see Figure 7). Knowing that the ratio between the small and great basis of the trapezoid representing the camera's view field is equal to 5 (for $L = 400cm$), the scaled area of scene synthesis verifies the same ratio as indicated by Figure 7)(right).

Let $b(x_b, y_b, z_b)$ be the barycenter of the box encompassing the object. The position of $b'(u_{b'}, v_{b'})$, the mapping of b on the synthesis area is calculated from the geometric relationship between the scene and the synthesis area as indicated by Figure 7. The values of $u_{b'}$ is equal to $d' \times x_b/d$ and $v_{b'}$ is equal to $d' \times (z_b - l)/d$ where d' is the small basis of the synthesis area. The values of l, L for the Kinect sensor are $80cm$ and $400cm$.

Now, let $b_1(x, y), b_2(x, y), .., b_n(x, y)$ be the points defining the box encompassing the object. The mapped points $b'_1(u, v), b'_2(u, v), .., b'_n(u, v)$ are located on the area. All pins inside the area defined by the points $b'_i, i = 1..n$ are set to level 2. The pins associated to the label of the object are set to their associated level 3, 4 or 5 at the barycenter as indicated by Figure 8 (Left).

The area's input may receive the 3D raw data or labels. In Figure 8 (Middle), the point cloud of the captured scene is directly mapped on the area. The same point could be mapped into the synthesis area using different heights of
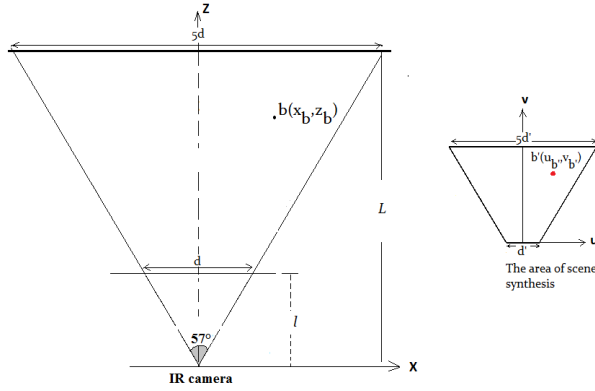
Fig. 7 (Left) The depth camera's view field. (Right) The area of scene synthesis where semantic labels are generated.
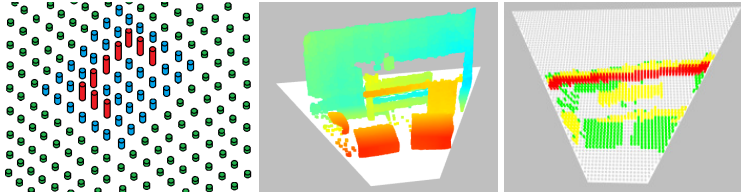

Fig. 8 (Left) The convex hull and the positioning of the label (here for a table) on the area. Note that the pins of the label in red are of level 3 (corresponding to the height of the table in the scene). The pins of the convex area in blue are of level 2. The remaining pins in the synthesis area (in green) are set to level 1 corresponding to the ground. (Middle) Mapping the 3D point cloud into the synthesis area. (Right) Scene synthesis from raw data (point cloud) using pins without semantic labeling.

pins (see Figure 8 (Right)). For more visibility, we assigned the grey color, the green color, the red yellow color and the red color to represent objects with level 2 (ground), 3, 4 and 5 respectively.

6 Experiments

6.1 Experimental environment and dataset

In order to perform the training, we ran our models on a GPU provided by the Google Colaboratory platform. After that, we executed our proposed system on a laptop having Intel Core $i5 - 7200U$ CPU ($2.50GHz \times 4$) and $4GB$ as processor and RAM respectively.

As to train our model, we extracted our 16 selected classes from Model-Net40 [19] (Figure 9), a public dataset which consists of 40 classes of CAD models. Note that we excluded doors and windows since they are represented by holes and in the real-world can be confused with noise or walls where they are closed. ModelNet40 originally includes 12311 models among which 2468
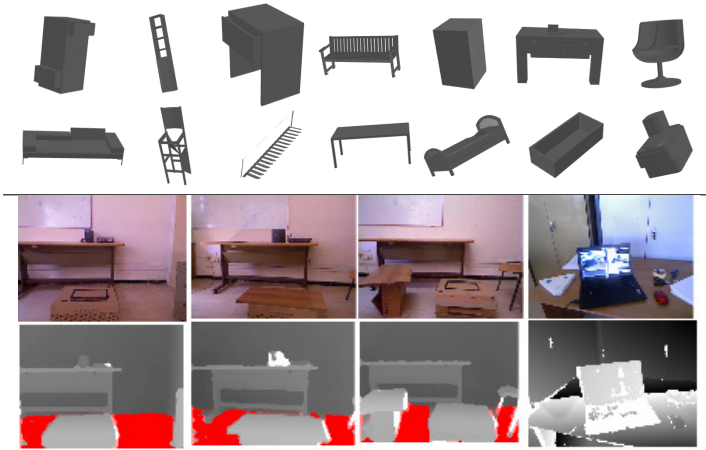
Fig. 9 Dataset samples. (TOP) Modelnet40[19]. (Down) GDIS[41]: the red color represents the detected ground.

are used for testing. Our constructed dataset includes 5560 models among 988 which are used for testing.

In addition, we used GDIS dataset (Ground Detection for Indoor Scenes) [41], our local dataset, for additional experiments. GDIS dataset was initially constructed for the ground detection task, but we considered it for point cloud classification and the validation of the entire system (from the ground detection to the semantic labeling).

6.2 Point cloud segmentation and feature extraction

Since we consider irregular segments and not planes, we computed the height as the height of the 90th percentile of the points' $y-coordinate$. We chose the 90th percentile instead of the 100th to avoid potential outliers. To evaluate this step, we took the measurements of some objects in the real-world and then compare it with the obtained geometric features. This is done by computing the Mean Absolute Error (MAE) between the objects' height and the computed one. The obtained results showed a slight mean difference that does not exceed $30mm$.

6.3 Semantic classification

Since PointNet is 3D raw data based, the CAD models are sampled into point clouds of 2048 points and normalized into a unit sphere. Regarding the experimental settings, we applied the same configurations as in [23]. As for data preprocessing, we, on-the-fly, applied random rotations around the yaw axis and jittered the points position as described in [23]. PointNet achieved the
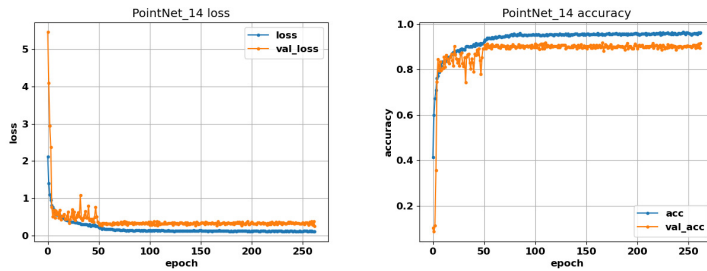
Fig. 10 (Left) PointNet loss function. (Right) PointNet accuracy. Training PointNet with 14 classes; the model starts to converge from epochs 50 and reach 96.19% and 91.67% on train-set and test-set respectively.

state-of-the-art with 89.2% accuracy in their original paper. In this work, we trained our model on a sub-set of 14 (after excluding doors and windows) classes from ModelNet40. The model started to converge from the 50th pechos and reached 91.67% accuracy on the test-set after 262 epochs (Figure 10).

However, as shown by the confusion matrix (see Figure 11), there is a high confusion between wardrobe and the classes bookshelf and dresser (about 20% and 10% respectively), between table and desk (about 16%), stool and chairs (about 15%). There is also a small confusion between the sofa and the classes bench and chair. These confusions are due to the geometric resemblance between them. This leads us to consider combining classes for the semantic classification as explained earlier in section 4. As the previous model, this model started to converge after the 50th epoch. The accuracy has been improved by 5% to reach 96.35% on the test-set after only 226 epochs (Figure 12). Thus, the described confusions have been eliminated (see Figure 13). The confusion matrix (see Figure 13 (Left)) shows that the confusion between the different classes does not exceed 7% unless for stairs that 15% of them were confused as chairs. To visualize this latter, we plot the three worst classification for each class (Figure 14): the instances have similar geometry with the predicted class with a certain probability. However, the confusion stills small and the system predicts accurately with high recall and precision (see Figure 13 (Right)).

The model was trained using 3D CAD models; on real-world data, the model predicts the right class when the input is noisy and slightly incomplete (see Figure 15); however, it fails otherwise. As shown in Figure 16 (column 2), the system has predicted incorrectly the class of the table that was poorly segmented (we only used background removal to simulate the cropped data). To overcome this problem, we only consider the predicted class with high probability (higher than 0.85).

Compared with the results of object classification based aid systems presented previously (Table 1), in the system [5] , authors didn't mention the accuracy of their proposed depth-based classification algorithm. As for the system [7], authors adapted FuseNet for RGB-D semantic segmentation that
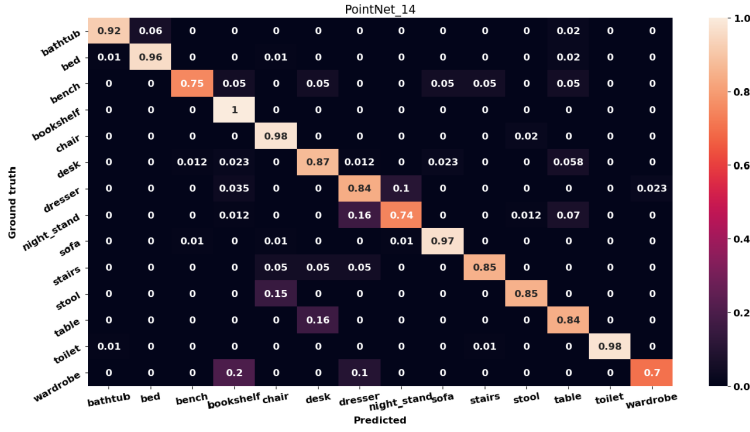
Fig. 11 Confusion matrix: there is high confusion between wardrobe and the classes bookshelf and dresser (about 20% and 10% respectively), between table and desk (about 16%), stool and chairs (about 15%); and small confusion between the sofa and the classes bench and chair.
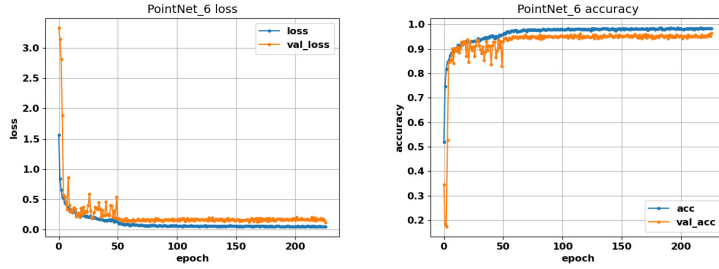

Fig. 12 (Left) PointNet loss function. (Right) PointNet accuracy. Training PointNet with 6 classes; the model starts to converge from epochs 50 and reach 98.26% and 96.35% on train-set and test-set respectively.

has 76.27% as accuracy. Compared to the third system wang2014segment, our model surpasses its accuracy and with a larger set of classes.

Table 1 Comparison with the classification module of the presented state of the art systems.

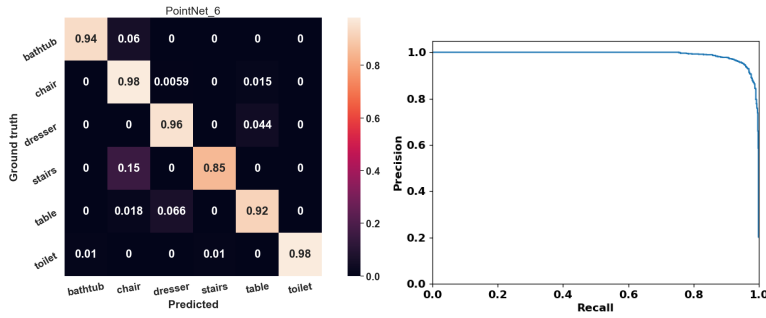|	Input	Nb Classes	Classifier	Accuracy (%)
Wang et al.[5]	Depth image	5	Depth-based classifier[5]	-
Lin et al.[7]	RGB-D	70	FuseNet[8]	-
Wang et al.[10]	RGB-D	4	Cascaded Decision Tree	71.45
Ours	Point cloud	14	PointNet[23]	91.67
Ours	Point cloud	6	PointNet[23]	96.35

Fig. 13 (Left) Confusion matrix: the confusion between objects does not exceed 6% unless for stairs that 15% of them were confused as chairs. Most of the objects are not confused with other objects (0% confusion). (Right) Recall-precision curve: the system predicts accurately with high recall and high precision).
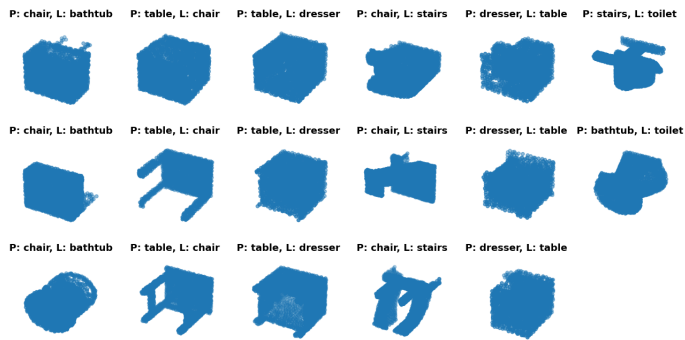

Fig. 14 Worst 3 classification results. P and L stand for Predicted and Label respectively. There is a geometric resemblance of the input instance with the objects of the predicted class.

6.4 Tests on GDIS dataset [41]

For more experiments using the entire system on real-world data, we show and discuss the output of each module for 4 images taken from a video sequence from our GDIS dataset [41] (Figure 17). After the ground detection (Figure 17 third row), we applied the DBSCAN clustering algorithm to break the occupied space into coarse segments (Figure 17 fourth row). After that, we compute for each segment its geometric features and its class. The Pointnet network has classified well the chair except for the last frame (Figure 17 last row, last column), it was classified as a table with a low probability (0.53), so the class was not mapped, only the convex hull was mapped. The model failed to predict the right class because the chair's point cloud was segmented into sub segments and this is due to the noisy nature of the Microsoft Kinect V1. If the class is not predicted, we set all the pins that represent the segment to the segment's level, rather than mapping the class to that level.
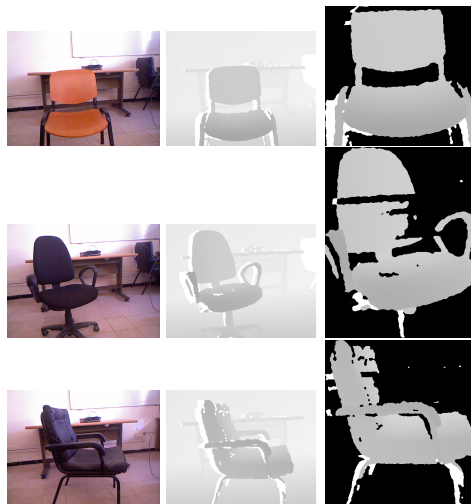
Fig. 15 From left to right: the captured scene (chair, chair, chair), its associated depth image (the system input), the resulting segment. The system has predicted: chair, chair, chair.

Fig. 16 From top to down: the captured scene (table, table, dresser), its associated depth image (the system input), the resulting segment. The system has predicted: table, chair and dresser.

7 Future Scope

In discussion section, we have mentioned two points that can be improved in the future respectively:

1. A second step can be added in the future for fine segmentation. In this step, objects that are arranged on a given coarse segment can be detected using coarse to fine segmentation. These latter, can be mapped on the synthesis
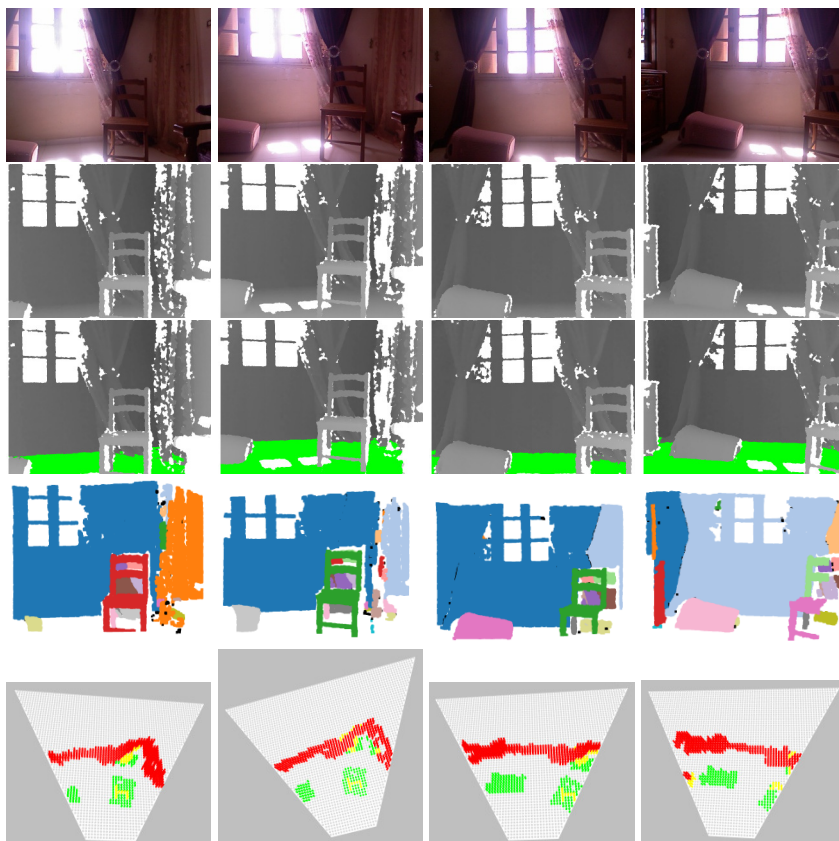
Fig. 17 (From Top) The RGB image, the depth image, the output of the ground detection algorithm, the segmentation using DBSCAN and the semantic labeling mapped on the synthesis area.

area only by commands from the user to avoid ambiguity while exploring the scene. In other terms, mapping coarse and fine information at the same time is time consuming and ambiguous such as mapping the table and the objects that are on it. We can imagine this scenario: the user is searching for a cup. After providing the global description, the user can locate the table on the synthesis area. After that, by long clicking on the table's label, we can at this time map on the whole surface of the area only the table and what's on it. At this step, by touching the new area's configuration, the user can have an idea about what is on the table and can search for his desired item.

2. A solution that we will consider in the future, is that the proposed model will be extended to work for a succession of frames and thus, the cropped objects can be completed by the alignment process and thus the prediction can be corrected.

On the other hand, we aim to cover other object classes. We also aim to provide a detailed semantic labeling, in a clear way, to include not only salient objects like tables, but also other small objects. This can be done by hierarchical classification by considering the classification of the salient objects. For example, to classify objects that are on the table the system will consider that the object is on the table.

8 Conclusion

In this paper, we proposed three main contributions:
-An object classification approach suitable for the visually impaired and blind people. It is based on deep learning networks implemented to classify the captured objects using a depth camera. Although we are using a state of the art neural network, we show that grouping object classes that have the same usage improves the system accuracy by 5%. This classification is combined with our previous approach [13] in order to extend the number of possible classes and to provide additional information. Using only a depth camera as an input sensor reduces the computational complexity and the system's cost in terms of the required hardware.
-A semantic labeling that is similar to the Braille and Kanji systems, can be mapped into a synthesis area. The proposed classification approach and its adequate semantic labeling will allow the user to have a clear idea about his surroundings and thus, he can perform several tasks alone.
-An end-to-end assistive system for the visually impaired and blind people. This system describes the captured scene architecture and provides geometric features and the nature of the detected obstacles mapped as touch-based semantic labels. This latter will help the user to understand his surroundings and thus, he can accomplish a set of tasks alone.

The proposed system fails to predict the semantic class of some point clouds due to the nature of real-world data that is incomplete and cropped. Different cropped objects can be similar to some cropped objects belonging to different classes. As current solution, the model ignores the predicted class if this latter has been predicted with a small probability.

References

1. Rabia Jafri, Syed Abid Ali, Hamid R Arabnia, and Shameem Fatima, Computer vision-based object recognition for the visually impaired in an indoors environment: a survey, in: The Visual Computer, 30(11):1197–1222, 2014.
2. M. Leo, G. Medioni, M. Trivedi, T. Kanade, G. Farinella, Computer vision for assistive technologies, Computer Vision and Image Understanding 154 (2017) 1–15.
3. T. R., B.Mocanu, T. Zaharia, Wearable assistive devices for visually impaired: A state of the art survey, Pattern Recognition Letters http://dx.doi.org/https://doi.org/10.1016/j.patrec.2018.10.031 doi:https://doi.org/10.1016/j.patrec.2018.10.031.
4. Sylvain Cardin, Daniel Thalmann, and Frédéric Vexo, A wearable system for mobility improvement of visually impaired people, in: The Visual Computer, 23(2):109–118, 2007.
5. H. Wang, R. K. Katzschmann, S. Teng, B. Araki, L. Giarré, D. Rus, Enabling independent navigation for visually impaired people through a wearable vision-based feedback system, in: 2017 IEEE international conference on robotics and automation (ICRA), IEEE, 2017, pp. 6533–6540.
6. A. Perez-Yus, D. Gutierrez-Gomez, G. Lopez-Nicolas, J. Guerrero, Stairs detection with odometry-aided traversal from a wearable rgb-d camera, Computer Vision and Image Understanding 154 (2017) 192–205.
7. Y. Lin, K. Wang, W. Yi, S. Lian, Deep learning based wearable assistive system for visually impaired people, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2019.
8. C. Hazirbas, L. Ma, C. Domokos, D. Cremers, Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture, in: Asian conference on computer vision, Springer, 2016, pp. 213–228.
9. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
10. Z. Wang, H. Liu, X. Wang, Y. Qian, Segment and label indoor scene based on rgb-d for the visually impaired, in: International Conference on Multimedia Modeling, Springer, 2014, pp. 449–460.
11. J. Bai, Z. Liu, Y. Lin, Y. Li, S. Lian, D. Liu, Wearable travel aid for environment perception and navigation of visually impaired people, Electronics 8 (6) (2019) 697.
12. J. Rivera-Rubio, K. Arulkumaran, H. Rishi, I. Alexiou, B. A.A., An assistive haptic interface for appearance-based indoor navigation, Computer Vision and Image Understanding 149 (2016) 126–145.
13. C. Zatout, S. Larabi, I. Mendili, S. A. E. Barnabé, Ego-semantic labeling of scene from depth image for visually impaired and blind people, in: 2019 IEEE/CVF International Conference on Computer Vision Workshops, ICCV Workshops 2019, Seoul, Korea (South), October 27-28, 2019, IEEE, 2019, pp. 4376–4384.
14. W. Jeamwatthanachai, M. Wald, G. Wills, Map data representation for indoor navigation by blind people, International Journal of Chaotic Computing 4 (1) (2017) 70–78.
15. J. Bai, S. Lian, Z. Liu, K. Wang, D. Liu, Smart guiding glasses for visually impaired people in indoor environment, IEEE Transactions on Consumer Electronics 63 (3) (2017) 258–266.
16. D. Maturana, S. Scherer, Voxnet: A 3d convolutional neural network for real-time object recognition, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 922–928.
17. C. Zatout, S. Larabi, A novel output device for visually impaired and blind people's aid systems, in: The First International Conference on Communications, Control Systems and Signal Processing, 2020.
18. S. Kumawat, S. Raman, Lp-3dcnn: Unveiling local phase in 3d convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4903–4912.
19. Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3d shapenets: A deep representation for volumetric shapes, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1912–1920.

20. Y. Liu, B. Fan, S. Xiang, C. Pan, Relation-shape convolutional neural network for point cloud analysis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8895–8904.
21. J. Li, B. M. Chen, G. Hee Lee, So-net: Self-organizing network for point cloud analysis, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 9397–9406.
22. C. R. Qi, L. Yi, H. Su, L. J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: Advances in neural information processing systems, 2017, pp. 5099–5108.
23. C. R. Qi, H. Su, K. Mo, L. J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 652–660.
24. J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, IEEE, 2009, pp. 248–255.
25. A. Zasimova, https://se.dreamstime.com/alfabet-braille-engelsk-version-f Alfabet braille (accessed February 15, 2020). https://se.dreamstime.com/alfabet-braille-engelsk-version-f.
26. J. Today, https://japantoday.com/category/features/opinions/why-you-must-learn-kanji Why you must learn kanji 2012 (accessed February 15, 2020). https://japantoday.com/category/features/opinions/why-you-must-learn-kanji
27. Y. Xie, J. Tian, X. X. Zhu, A review of point cloud semantic segmentation, arXiv preprint arXiv:1908.08854.
28. R. Kuçak, E. Özdemir, S. Erol, The segmentation of point clouds with k-means and ann (artifical neural network), The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 42 (2017) 595.
29. S. Aparajithan, S. Jie, Clustering based planar roof extraction from lidar data, in: ASPRS 2006 Annual Conference, 2006.
30. T. Melzer, Non-parametric segmentation of als point clouds using mean shift, Journal of Applied Geodesy Jag 1 (3) (2007) 159–170.
31. J. M. Biosca, J. L. Lerma, Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods, ISPRS Journal of Photogrammetry and Remote Sensing 63 (1) (2008) 84–98.
32. T. Czerniawski, B. Sankaran, M. Nahangi, C. Haas, F. Leite, 6d dbscan-based segmentation of building point clouds for planar object classification, Automation in Construction 88 (2018) 44–58.
33. B. Wu, A. Wan, X. Yue, K. Keutzer, Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 1887–1893.
34. C. Wang, M. Ji, J. Wang, W. Wen, T. Li, Y. Sun, An improved dbscan method for lidar data segmentation with automatic eps estimation, Sensors 19 (1) (2019) 172.
35. A. Krizhevsky, One weird trick for parallelizing convolutional neural networks, arXiv preprint arXiv:1404.5997.
36. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.
37. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
38. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.
39. R. Cupec, I. Vidovic, D. Filko , P. Durovic. Object recognition based on convex hull alignment. Pattern Recognition 102 (2020), DOI: 107199.
40. M. Ester, H.P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise., in: Kdd, Vol. 96, 1996, pp. 226-231.
41. C. Zatout and S. Larabi. Dataset of ground detection for indoor scenes (GDIS dataset). http://perso.usthb.dz/ slarabi/DGIS.html, 2019

42. E. Schubert, J. Sander, M. Ester, H. P. Kriegel, X. Xu, Dbscan revisited: why and how you should (still) use dbscan, ACM Transactions on Database Systems (TODS) 42(3), (2017) 1-21.

43. Stefan Milz, Martin Simon, Kai Fischer, Maximillian Pöpperl, and Horst-Michael Gross, Points2pix: 3d point-cloud to image translation using conditional gans, in: German Conference on Pattern Recognition, pages 387–400, Springer, 2019.

44. Sameera Ramasinghe, Salman Khan, Nick Barnes, and Stephen Gould, Spectral-gans for high-resolution 3d point-cloud generation, arXiv preprint arXiv:1912.01800, 2019.

Author biographies

Chayma Zatout, is a second year PhD student in Artificial RIIMA Laboratory at USTHB University, Algeria. Her doctoral research investigates the use of 3D data to generate a semantic labelling for scene understanding. She holds a master's degree in Intelligent Computer Systems from USTHB University.

Slimane LARABI, received Ph.D. in Computer Science in 1991 from the I.N.P. of Toulouse, France. From 1992 to 2007, he was an assistant professor at USTHB University of Algiers. Since 2007, he is professor at the same university and leads research in Computer Visual Computing Group of the RIIMA Laboratory. His current research topics are scene reconstruction, recognition, semantic coding and visual positioning.