

AMERICAN UNIVERSITY OF BEIRUT

OBJECT DETECTION CONSTRAINED BY
ONTOLOGICAL PRIORS

by

GEORGES ANTOINE CHAHINE

A thesis

submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Mechanical Engineering
of the Faculty of Engineering and Architecture
at the American University of Beirut

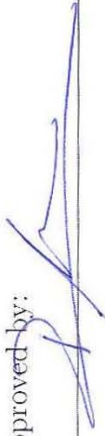
Beirut, Lebanon
December 2016

AMERICAN UNIVERSITY OF BEIRUT

OBJECT DETECTION CONSTRAINED BY
ONTOLOGICAL PRIORS

by
GEORGES ANTOINE CHAHINE

Approved by:



Dr. Daniel Asmar, Associate Professor
Mechanical Engineering

Advisor



Dr. Najib Metni, Associate Professor
Mechanical Engineering

Member of Committee



Dr. Elie Shammam, Assistant Professor
Mechanical Engineering

Member of Committee

Date of thesis defense: December 21, 2016

AMERICAN UNIVERSITY OF BEIRUT

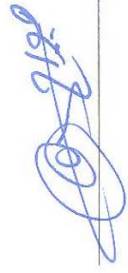
THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Chahine Georges Antoine
Last First Middle

Master's Thesis Master's Project Doctoral Dissertation

I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One** ---- year from the date of submission of my thesis, dissertation, or project.
Two ---- years from the date of submission of my thesis, dissertation, or project.
Three ---- years from the date of submission of my thesis, dissertation, or project.

 _____
Signature Date
27/12/2016

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgments

This work was supported by the Lebanese National Council for Scientific Research (LNCSR) and the University Research Board (URB) at the American University of Beirut. I would like to extend my gratitude to family, faculty, friends and colleagues for all the technical and emotional support.

An Abstract of the Thesis of

Georges Antoine Chahine for Master of Engineering
Major: Mechanical Engineering

Title: Object Detection Constrained by Ontological Priors

The problem of object detection in Computer Vision is a difficult and interesting problem which is far from being solved due in no small part to the challenges of perception. Nevertheless, by introducing top-down priors such as semantics, the problem of segmenting and detecting objects becomes traceable. This paper proposes such an approach by relying on the ontological relationships that make up parts of objects in order to enhance their detection.

The proposed method processes the point cloud of a scene and clusters it into pools of potential objects. Hypotheses on the object identity is generated using geometric and customized ontological definitions to generate probabilistic models that would constitute the building blocks for the decision making process. An object labeling scheme derived by minimizing an energy function is presented. Finally, objects are replaced by matching them to generic CAD models.

To evaluate the proposed method, we run our experiments on three well-known datasets and compare with results in the literature. Results show superiority to the prior art in terms of both recall and precision.

Contents

Acknowledgements	v
Abstract	vi
1 Introduction	1
2 Literature Review	4
2.1 Thresholding Methods	4
2.2 Other Feature Based Methods	5
2.3 Machine Learning	5
2.4 Semantic Labeling and Ontology	6
3 Object Detection System	9
3.1 Concerning Ontology	9
3.2 Point Cloud Generation	10
3.3 Filtering, Reconstruction, Clustering and Plane Segmentation	10
3.4 Probabilistic Framework for Object Detection	11
3.5 Energy Function Optimization	15

3.6	Developing the General Probability Related to the Existence of an Object	16
3.7	Generic CAD Model Replacement	18
4	EXPERIMENTS AND RESULTS	20
4.1	Setup and Datasets	20
4.2	Results and Discussion	21
5	Conclusion	39
A	Abbreviations	40

List of Figures

1.1	The Clearpath Husky robot in our vision and robotics lab equipped with a monocular camera is used during experimentation. The 2D lasers on the robot were not used in the experiment.	2
2.1	Regression, whether linear or not, is a example of learning a trend from numerous amount of data. [14]	6
3.1	System overview.	11
3.2	Sample constraints extracted from the ontology of a chair	12
3.3	The combination of top-down information along with information extracted from the point cloud is fed into a probabilistic decision making process.	13
4.1	Sample ontology flow chart for a seating area. (Figure reproduced from [19]	22
4.2	Scene from the vision and robotics lab showing a chair being detected by probabilisitic ontology.	23
4.3	Home scene from the dataset of Koppula et al. [1] showing successful detection of a sofa.	24
4.4	Office scene from the dataset of Koppula et al. [1] showing successful detection of a partially occluded chair and a false positive consisting of table desk.	24

4.5	Statistical measure of the prevalence of seating areas in urban outdoors. The figure shows robust stabilization as of sample 78. The horizontal axis reflect the number of processed while the vertical axis shows the value of $P_{obj/out}$	25
4.6	Statistical measure of the prevalence of seating areas found indoors in a random scene. The figure shows robust stabilization as of sample 49. The horizontal axis reflect the number of processed while the vertical axis shows the value of $P_{obj/in}$	25
4.7	Sample Data Sampling for 4 out of 24 indoor categories. The horizontal axis represent the number of samples, whereas the vertical axis represent the probability of finding a seating place, on a scale from 0 to 1.	34
4.8	Left to right, the figure shows the original scene consisting of my own living room, followed by the captured point cloud using Visual SFM [2]. The second row features a map of detected objects by the same map, overlapped with CAD models. The bottom row features the final generic map, followed by the same map with color information restored according to the original point cloud colors.	36
4.9	Skeleton of a chair, used to match with the point cloud of the detected chair.	37
4.10	The transformation found by matching the skeleton to the object is then applied to the model in the figure above.	37

List of Tables

3.1	Type of Environment	17
4.1	System Parameters	23
4.2	Sample Cluster Probabilities of Fig. 4.4, Office Scene	26
4.3	Comparison with Literature, Seminar Room Dataset	26
4.4	Comparison with Literature, Office Scenes Dataset	27
4.5	Comparison with Literature, Home Scenes Dataset	27
4.6	Comparison with Literature, Seminar Room Dataset	27
4.7	Comparison with Literature, Office Scenes Dataset	28
4.8	Comparison with Literature, Home Scenes Dataset	28
4.9	Cost of Assigning a label l to a Cluster Consisting of a Table	28
4.10	Probability of Finding a Seating Area	30
4.11	Environment Classification P_N for Office Scenes	31
4.12	Environment Classification P_N for Home Scenes	32
4.13	Environment Classification P_N for Seminar Rooms	33
4.14	Comparison with Literature, Seminar Room Dataset	34
4.15	Comparison with Literature, Office Scenes Dataset	34

4.16 Comparison with Literature, Home Scenes Dataset	35
--	----

Chapter 1

Introduction

Scene understanding is an active field of research, spanning several challenging topics in computer vision, from segmentation, to object detection, and object recognition. The central challenge facing these problem is that of perception, where a scene may lend itself to different interpretations. Recent developments in machine learning algorithms [3] showed good results; however, the methods require enormous numbers of training samples and are found to be vulnerable to changes in aspect ratio and occlusions.[4].

Alternatively, by taking advantage of top-down information, such as matching to CAD models [5], the solution space for segmentation, detection, and recognition becomes more constrained; thereby yielding systems that are more reliable and more consistent. The use of ontology [6], has recently also seen its fair share in computer vision, being used to analyze the relationship between different parts of an object in order to identify the nature of that object. The concept of ontology was first suggested by Parmenides in the sixth century BCE [7], whereby any object can be described by its constituents, the geometric relationships between them, as well as the interaction of that object with its environment.

In this paper, we make use of this definition to establish geometric and topological relationships between parts of an object in order to produce strong priors that help build parts of objects into complete units. Furthermore, the concept of ontology in computer vision is formalized, for the first time, through a probabilistic mathematical model. Subsequently, segmentation and detection is



Figure 1.1: The Clearpath Husky robot in our vision and robotics lab equipped with a monocular camera is used during experimentation. The 2D lasers on the robot were not used in the experiment.

performed by minimizing an energy (cost) problem. Thereafter, the final map is generated by replacing detected objects with generic objects. This is done through matching of point clouds with geometrical constraints, detailed in Chapter 3.

Our proposed system is capable of detecting and localizing objects regardless of the environment. The system includes a probabilistic ontological model and takes as input a point cloud to generate object class for each cluster. Subsequently, object selection is performed by minimizing an energy function.

To validate our work, experiments were performed on three datasets; namely, the Furniture Recognition Dataset [5], and two other datasets extracted from the Cornell-RGBD-Dataset [8] and used by [1]. In our experimental section we benchmark our results to the prior art using these three datasets.

The innovation presented in this work consists of a probabilistic framework for object detection using ontology, a scheme to predict the existence of an object using statistics and machine learning, refined by minimizing a cost function.

The remainder of this thesis is structured as follows. Chapter 3 details the specifics of our proposed system from the extraction of geometric primitives to the interpretation of ontological definitions in objects, to the decision making process. Experiments and results are presented in Chapter 4 and finally we conclude the thesis in Chapter 5.

Chapter 2

Literature Review

Object detection is an old yet trending topic in Computer Vision. The reason for that is the ever evolving needs for reliable techniques constantly outperformed by newer and more reliable innovations. So far never perfected, object detection has become a multi-disciplinary field attracting engineers and computer scientists alike, by combining robotic intelligence with evolving algorithms and technologies.

Being a well established perception problem, object detection has evolved from simple color thresholding methods to the most recent active perception methods using semantic labeling and/or ontology [9].

2.1 Thresholding Methods

Thresholding segmentation [10] is the simplest form of image segmentation. For example, black and white segmentation is performed by directly measuring pixel intensity against a threshold. Other forms of thresholding methods include color segmentation, entropy segmentation to separate background from foreground information, and histogram methods by detecting and analyzing data peaks and curvatures. Smoothness constraints are often applied to improve clustering quality.

2.2 Other Feature Based Methods

Feature Based methods heavily rely on feature extraction methods such as color extraction or edge detection. The canny edge detector [11] is an example of feature extraction methods, often conjunctively used with hough line transform to detect object shape by drawing its contour. Feature extraction methods constitute the building blocks of several Simultaneous Localization and Mapping (SLAM) algorithms such as ORB SLAM [12].

Other types of feature extraction methods include SIFT and GIST [13] descriptors, often labeled as low level segmentation systems, they are widely used with machine learning algorithms to fill the gap between real world data and computerized information.

2.3 Machine Learning

Machine learning, at its simplest form, has been used for decades in order to plan daily life and to try to scientifically predict the future. For instance, factories monitor consumer behavioral patterns and try to predict demand by learning and trying to predict future consumer behavior. Linear Regression, extrapolation and other statistical models are all simple forms of computerized prediction algorithms.

Learning algorithms are roughly divided into three categories: Supervised, Unsupervised and Reinforcement Learning [15]. The most trivial type of machine learning is Supervised Learning, whereas a multi-variable classification function is estimated. The variable are hence estimated by providing samples with known ground truth. The amount of training data required depends on the minimum acceptable accuracy therefore, the training phase is often completed on several steps, with the classifier accuracy being tested after each training step is completed. This process persists till the classifier reaches satisfactory prediction rates.

Unsupervised learning algorithms are mostly used in clustering methods. An example of which is the K-means algorithm, taking as input the required number of clusters and accordingly maps the data into the specified number of pools.

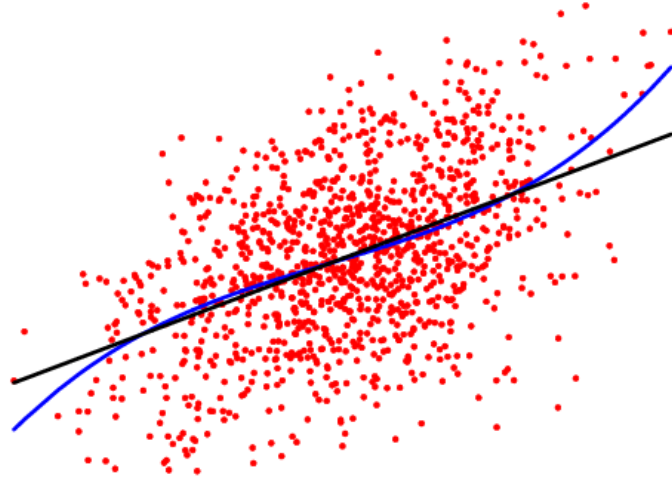


Figure 2.1: Regression, whether linear or not, is an example of learning a trend from numerous amount of data. [14]

Reinforcement Learning algorithms is resemblant to Supervised Learning, with the exception that the training phase is automatic, mostly by trial and error. The machine is therefore continuously exposed to an environment while learning from past experiences in a closed loop fashion.

One interesting contribution by Pillai et al. [16] takes advantage of machine learning while feeding on monocular camera stream to develop a new SLAM-aware object recognition system. The system takes advantage of different angles and viewpoints along with a trained classifier in order to achieve an object recognition system that is more reliable than doing so using a frame by frame approach, the latter leading to more false positives.

2.4 Semantic Labeling and Ontology

Semantic Labeling and Ontology are often discussed in the same context in field of object detection. Ontology is the easiest concept to understand since it imitates the human way of thinking. For instance, a table could be identified by 4 legs holding an horizontal plane: the logic used to identify the table is the same one used by the human way of thinking, as well as the computerized version of the same table. Another example is detecting the lower shelf of a three-story

shelve: A human will identify the lower shelf by verifying the existence of 2 shelves above the considered shelve. Same goes for the below code in OWL-DL standard, to detect the lower shelf plane:

$$\begin{aligned} \textit{Shelf} = & \textit{LowShelfPlane} \textit{and} \\ & (\textit{isBelowsome}(\textit{MiddleShelfPlane} \textit{and} \\ & (\textit{isBelowsome} \textit{HighShelfPlane}))) \end{aligned}$$

Therefore, it is hypothetically sound to say that any object that can be recognized by humans can also be identified using ontology by following an equivalent strategy to describe the existence of an object. This concept has also been further pushed into detecting the ontology of an event, otherwise known as domain ontology [17], an example of which is the automatic detection of a car crossing a red light using video surveillance footage.

Semantic Labeling [18] is the process of assigning a label to an object by evaluating the conformity of the object to the label requirement. This is usually done through a cost optimization process, whereas the goal is to assign to an object the label that incurs the lowest cost. Smoothness constraints are employed to ensure homogeneous object labeling through pixels or voxels.

Selvatici et al. [6] [19] used semantic labels for objects with exactly known identities along with structure from motion techniques in order to reconstruct from a monocular camera the environment into a map of objects. Similarly, Mason et al. [7] [20] presented another deterministic semantic approach to mapping; point clouds are used for both plane fitting and for semantic feature extraction, leading up to a semantic map at the level of objects. The algorithm is also able to detect object position change. All these methods introduce interesting innovation but do not afford uncertainty in their decision making process. The added value of relying on a probabilistic framework is to account for uncertainties through a tolerant decision making process, supported by an optimization step where the resulting object detection is refined. All of this is confirmed through improved results, as presented in Section III.

More closely related to our method is the recent work of Gnther et al. [3] [5], where their system offers a method for building semantic object maps of furniture, it can work on any point cloud and uses CAD models along with the Iterative Closet Point (ICP) algorithm to build a map of objects. Koppula et al. [8] [1] specializes in detecting objects in large cluttered rooms also using semantic labels. Both papers use deterministic ontological methods for which we saw an

opportunity to improve by treating the problem from a probabilistic perspective.

Up so far and to the best of our knowledge, there has been no significant contribution aimed at detecting objects using probabilistic ontology or similar approach. We firmly believe that object can be greatly beneficial. For instance, people with special needs can be guided by robots to an eligible seating place, detected by the guiding robot. Other possible applications include mapping and landmark identification, estimation of seating capacity for big rooms or outdoor venues using an Unmanned Aerial Vehicle and with some code modifications it can be used for identification of distinct geometrical shapes for military applications.

Salas-Moreno et al. [21] developed a new 3D SLAM algorithm also known as SLAM++ that uses knowledge of commonly available objects inside an environment to produce a map of objects. Similarly to [22], this method is however limited to the availability of a depth map, usually through a Kinect camera.

Chapter 3

Object Detection System

3.1 Concerning Ontology

Detecting objects of a known nature brings several advantages to the research problem, through the possibility of adding top-down information to the problem. Object detection, as shown in Fig. 1, is where the hypotheses about the nature of an object is generated using top-down probabilistic ontological priors combined with bottom-up extracted geometrical information. Ontology, as previously introduced, is the systematic study of the existence of an object by analyzing the parts that form it and the relationships between them. As a simple example consider a chair made up as a seating area for one person, a back support for the seated person, and a support for the seating area to the ground (i.e., legs).

Examples of relationships between these parts (as shown in Fig. 2) include proprioceptive ones such as the adjacency of the seating area to both the back support and the legs, and the approximately normal angle between the seating area and these supports. Relationships could also be of an exteroceptive nature such as the height of the seating area above ground, and the contact of the legs with the ground plane.

In this paper the proposed system detects objects by relying on such ontological relationships, while accounting for measurement inaccuracies and shape

inconsistencies, through a probabilistic model taking as input both point cloud and ontological constraints, and generating probabilistic inference on the nature of an object.

3.2 Point Cloud Generation

Different sensors can be used to extract a 3D points cloud from a scene. Options include LiDARS, Kinect sensors, and stereo cameras. In addition, monocular cameras can also be used for the same purpose by performing Simultaneous Localization and Mapping (SLAM), although the resulting maps, which are obtained are to an unknown scale. There are various different flavors of Visual SLAM implementations, each possessing different advantages and disadvantages in terms of tracking and point cloud quality [23]. Of the available systems proposed in the literature, LSD SLAM typically produces the most dense point clouds; for this advantage it is favorable to segmentation methods applied to point clouds.

Since the maps resulting from monocular SLAM are to an unknown scale, several options are available to recover this information; for instance, it is typical to pair the monocular camera with an Inertial Measurement Unit (IMU), or equivalently as in our case, recovering the ego motion of the camera by fusing robot supported encoder measurements, and use the corresponding differential scaled motion to correct the map of the SLAM map [24]: $X_sc = k_s X_c$, where X_c is the input point cloud, k_s the scale inferred from encoder feedback and X_sc the scaled point cloud. $k_s = D_e/D_s$, and D_e, D_s represent the distances traveled according to the encoders and monocular feedback, respectively.

3.3 Filtering, Reconstruction, Clustering and Plane Segmentation

Filtering is applied early to each point cloud in order to remove noise. This step includes both de-noising and the removal of far outliers caused by measurement errors. Subsequently, voxelization is performed whereas points are presented as voxels with a finite volume, in order to reduce point cloud size and significantly reduce the processing power required in subsequent steps. Meshing through surface reconstruction as well as several graph optimization runs are subsequently

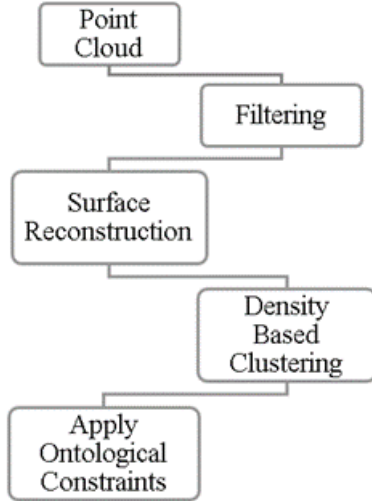


Figure 3.1: System overview.

performed to attain smoother surfaces and reduce scattering through plane fitting, subject to smoothness constraints. Clustering, blobbing or object extraction is the process of extracting candidate object clusters, which are subsequently analyzed to identify their identity. This is done by grouping neighboring points into candidate clusters. To end with, Planes are extracted from clustered objects using the M-estimator Sample Consensus algorithm (MSAC) [25], a variant of the Random Sample Consensus (RANSAC) [26].

3.4 Probabilistic Framework for Object Detection

We present the below probabilistic ontological framework for the proposed system:

$$P_{obj/ont} \propto P_{obj}P_{ont/obj} \tag{3.1}$$

Where $P_{obj/ont}$ is the maximized hypotheses probability generated over the identity of the object, $P_{ont/obj}$ is modeled as a product of independent Gaussian distributions, thereby allowing the expression of the total conditional probability

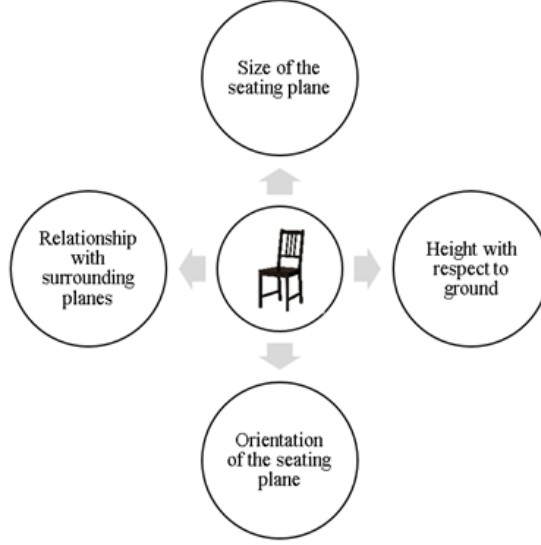


Figure 3.2: Sample constraints extracted from the ontology of a chair

as a product:

$$P_{ont/obj} = \Pi P_{def} \quad (3.2)$$

P_{Def} is the normal distribution [27] $N(\mu, \sigma^2)$ inferred from our ontological definitions:

Here, x is the information extracted from the point cloud such as plane orientation, the mean and the variance are set according to the ontological priors. The constituents are therefore presented as follows:

$$P_{def} = P_{Pr} P_{Or} P_{De} P_{He} \quad (3.3)$$

P_{Pr} , the prior probability relating to the geometrical relationship of a plane with the surrounding planes in the same cluster including constraints on plane size. A cluster containing planes of the correct size and forming the correct angles between them will score higher values for this probability.

P_{Or} , the prior related to a part orientation; for example, in the case of a chair, it could be the relative orientation between seating area and back support, set using ergonomic standards [28]. The constraints governing this prior will

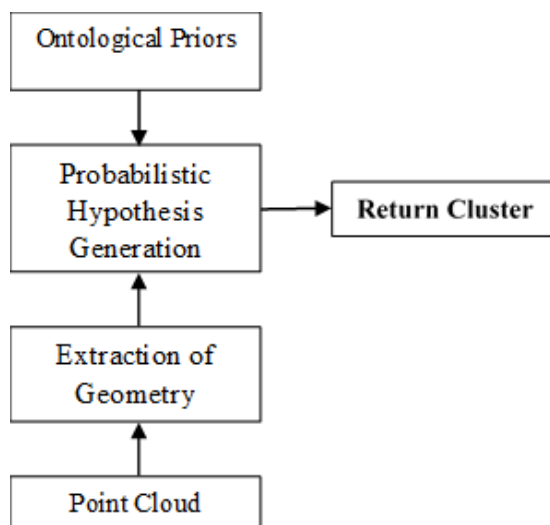


Figure 3.3: The combination of top-down information along with information extracted from the point cloud is fed into a probabilistic decision making process.

usually favor planes with orientations favorable to those of a known object.

P_{De} , is the prior related to point cloud density i.e., the existence of a plane. Even though extremely small clusters are being disregarded at the start, false positives are recurrently associated with lack of information in the point cloud. Adding this probability will account for poorly represented clusters in the probabilistic model.

P_{He} , the probability prior related to part height or position inside the point cloud; in the case of a chair, it can be set using ergonomic standards [28]. This prior provides qualitative value to the model by constraining one or more variables related to cluster location, i.e., objects found at a certain altitude from the Ground or other objects more likely to be found in corners such as in the case of garbage bins.

P_{obj} is the general existence of an object and is closely related to the choice of the detected subject for instance, a chair in a given scene or point cloud is more abundant indoors than outdoors. The below breakdown of P_{obj} into indoors and outdoors is attributed to the chair example presented in Section III. At a performance cost, P_{obj} can be altogether eliminated whenever it is not possible to estimate the variable (such is the case when the likelihood of finding an object is equal everywhere) by setting its value equal to 1.

$$P_{obj} = P_{objnin} + P_{objnout} \quad (3.4)$$

Developing the above equation yields:

$$P_{obj} = P_{in}P_{obj/in} + P_{out}P_{obj/out} \quad (3.5)$$

where P_{in} and P_{out} represent the probabilities of being indoors and outdoors, respectively. $P_{obj/in}$ or $P_{obj/out}$ is the general probability of finding a given object indoors or outdoors, respectively. A valid estimation of the environment (P_{in} and P_{out}) was achieved using a Support Vector Machine (SVM). The pseudo code of the implementation is shown in Algorithm 1.

Algorithm 1 Probabilistic Ontology

Input: Point Cloud M

Output: Detected Clusters

```

1: Denoise  $M$ 
2: Reconstruct  $M$ 
3: Voxelize  $M$ 
4: Downsample  $M$ 
5: for every point  $(x, y, z)$  in  $M$  do
6:   Compute Normals
7:   Cluster Nearby Points
8:   Fit Planes using MSAC
9: end for
10: for every plane  $i$  in cluster  $j$  do
11:   Apply Ontological Constraints  $P_{Def-i}$ 
12:    $P_{i-j} = P_{obj} \Pi P_{Def-i}$ 
13: end for
14: for every cluster  $j$  do
15:    $P_j = \max(P_{i-j})$ 
16:   if  $p_j > threshold$  then return Cluster
17:   end if
18: end for

```

3.5 Energy Function Optimization

As we obtain the hypotheses for object class from probabilistic ontology (Algorithm 1), we build an energy function given object class and formulate the energy function E as follows:

$$E = \sum_{c_i} U(s_i, l_i, x) \quad (3.6)$$

where the goal is to assign a label $l_i = l_1, l_2, \dots, l$ to each cluster c_i . U represents both the costs of assigning class s_i to cluster c_i as well as the cost of assigning label l_i to class s_i . Using the probabilities generated in (2), we develop the term U as follows:

$$U = 1 - \beta / (1 + e^{-p}) - \gamma / (1 + e^{-f(c_i, l_i)}) \quad (3.7)$$

where β and γ are weights to balance the cost of assigning an object class to the cost of assigning a label such that $\beta + \gamma = 1$. The first term is directly inferred from the previous ontological relationships, whereas the second term is a discriminatory term that penalizes label assignment according to labeling constraints. The function $f(c_i, s_i, l_i)$ is modeled as a product of independent Gaussian distributions taking as input object class and cluster information to measure conformity of each label to the assigned cluster:

$$f(c_i, l_i) = \prod_{A_k=1|c_i}^{A_n} F_{I|l_i} F_{m|l_i} \quad (3.8)$$

where F_I and F_m are Gaussians as shown previously in (4), related to object color and plane size, respectively. For a given label l_i , $F_{I|l_i} F_{m|l_i}$ is calculated for every plane A_k inside the cluster c_i .

3.6 Developing the General Probability Related to the Existence of an Object

In the previous section we have shown how the breakdown of P_{obj} can help the system identify an object. In this section, we furthermore explore the possibilities associated with the variable P_{obj} .

It is believed that P_{obj} can play a bigger role, more like a wild card in the decision making process. We previously broke P_{obj} into indoors and outdoors in the example of a chair. We now present a generalized proposal for indoor estimation of P_{obj} :

$$P_{obj} = P_{obj \cap N_1} + P_{obj \cap N_2} + P_{obj \cap N_3} + \dots + P_{obj \cap N_{24}} \quad (3.9)$$

$$P_{obj} = P_{N_1} P_{obj/N_1} + P_{N_2} P_{obj/N_2} + P_{N_3} P_{obj/N_3} + \dots + P_{N_{24}} P_{obj/N_{24}} \quad (3.10)$$

Hence we have assumed that the indoor world is divided into twenty-four categories, enlisted from the MIT places205 dataset in Table 3.1.

$P_{obj/N_1}, P_{obj/N_2}, \dots, P_{obj/N_{24}}$ are estimated using Stratified Sampling.

$P_{N_1}, P_{N_2}, \dots, P_{N_{24}}$ are estimated using a support vector classifier machine (CSVM). The latter methodology is furthermore discussed in Chapter 4.

In contrast to dividing the environment into indoors and outdoors, we have presented a generalized method for the estimation of P_{obj} . The only disadvantage presented with this estimation is the effort required to determine, through stratified sampling, the environments for which the object is mostly to be found: This implicit step is essential especially if the robot is expected to navigate through different rooms. The final product of the above equations will directly affect the cost of assigning a given label to the object. Since the decision making process is done through cost minimization, the variable P_{obj} will play a discriminatory role by increasing the cost of having an object in an unfavorable environment.

Table 3.1: Type of Environment

Variable	Reference	Available Data
N_1	Airport Terminal	15100
N_2	Auditorium	15100
N_3	Cafeteria	5184
N_4	Classroom	15100
N_5	Conference Room	9154
N_6	Corridor	15100
N_7	Dining Room	15000
N_8	Dorm Room	6202
N_9	Food Court	7361
N_{10}	Home Office	13942
N_{11}	Hotel Room	15100
N_{12}	Kindergarden Classroom	6395
N_{13}	Kitchen	15100
N_{14}	Kitchenette	15100
N_{15}	Living Room	15100
N_{16}	Lobby	15100
N_{17}	Museum	7919
N_{18}	Music Studio	15100
N_{19}	Office	15120
N_{20}	Reception	7311
N_{21}	Shoe Shop	5184
N_{22}	Staircase	15100
N_{23}	Television Studio	5940
N_{24}	Waiting Room ₁₇	5921

3.7 Generic CAD Model Replacement

At this stage, objects along with appropriate labels are identified however, it would be beneficial to be able to visualize detected objects in a generic form. Since ontology imitates the human way of thinking, we decided to proceed with generic CAD models, whereas the purpose here is attempting to imitate human visual memory. For instance, most people would not remember the exact shape of objects after visiting a room for the first time, especially color variations and curvatures. Even-though CAD model replacement has been considered before in the prior art, it usually consisted of acquiring the exact models of detected objects by scanning, drawing or contacting the manufacturing company. Overcoming the challenges of generic model replacement holds therefore an important advantage in terms of convenience. After reviewing methods used in history, we discuss the challenges and present our own approach to the research problem.

The challenges that lay behind generic CAD model replacement differ from one class of objects to another. The below example are sample challenges associated with object class "seating area":

- Variations in color
- Different seating plane size
- Absence or Presence of arm rests
- Different configurations chair support, such as legs.
- Inconsistency of captured point clouds
- Similarity in color for closely spaced objects

Below are some of the proposed solutions for the same challenges above:

- Removal of color information from the decision making process
- This can be overcome by including several options for generic CAD models. The chosen CAD model is the one holding the highest match percentage or the smallest RMS error. Future work might include generating a generic CAD model that fits the detected object's dimensions.

- Include CAD generic arm rests with limited thickness to reduce the chance of false association with other parts of the model./
- Same as above.
- Use of Grid-Average down-sampling methods.
- Use of other clustering techniques, such as region-growing

CAD model fitting is achieved by finding the transformation matrix that best fits the model to the detected object. To do so, we used the Iterative Closest Point algorithm (ICP) to retrieve the affine transformation. Subsequently, the transformation is applied to the chosen CAD model according to object label. Color information is finally restored by taking the average object color and applying it to transformed CAD models. The final map is the collection of all the transformed CAD models grouped into a single point cloud.

By taking the assumption that all objects are standing still on the ground in their upright position, we can constrain all movement in the vertical direction, as well as constraining both roll and tilt angles. Even-though using a two-dimensional version of ICP was considered, it would incur a valuable loss of information in the vertical direction therefore, we chose to recover the yaw angle through the recovery of the Euler angles from the three-dimensional rotation matrix given through ICP. Consequently, the resulting model only allows CAD models to move while connected to the ground at all times, with flexible orientation.

Chapter 4

EXPERIMENTS AND RESULTS

4.1 Setup and Datasets

In order to assess the proposed system, we set and customize our ontological priors to match those of a seating area. We will also assign three possible labels l_1, l_2 and l_3 to the class seating place, namely regular chair, sofa, and table, respectively. As most false positives consisted of tables, the third label was added to the class since the ontologies of tables and chairs have overlapping constituents, such as legs and horizontal planes that can be seated on.

Measurements higher than 1.4 meters above Ground level were disregarded in order to reduce chances of false positives, as no seat or table could exist at such height. Points near each other will be clustered together through density based clustering [29]. As for surface reconstruction and voxelization, we use the Las Vegas Reconstruction (LVR) toolkits marching cubes algorithm for both its reliability and its proven superiority in literature as shown in [5] [30] and [31]. The full list of parameters for our system is shown in Table 4.1.

The proposed system is first tested in one of our labs and then benchmarked against the state of the art using three well-known datasets. For the first experiment, held at our VRL lab, we used a Clearpath Husky robot (shown in Fig. 1.1), equipped with a Point Grey monocular camera. Using the encoder feedback

for the mobile robot, along with the correct transformation from robot frame to camera frame, monocular SLAM is initialized with correct scale. The computer in use is a regular fourth generation Intel Core i7, equipped with 10 GB of RAM and an Nvidia graphics card. The main code is written in Matlab, with the exception of data capture using ORB SLAM, surface reconstruction and 6D SLAM: these were on Ubuntu 14 Trusty, with ROS indigo installed.

The first dataset from literature is the seminar room used by Gnther et al. [5] The dataset consists of over 378 RGBD point clouds registered using 3DTK format. We stitch the map using 3DTKs 6DSLAM [32] and experiment with our system in ten consecutive runs. Surface reconstruction, voxelization and several runs of optimization were performed using the Las Vegas Reconstruction Toolkit [31].

The second [1] and third datasets [8] consisted of 28 and 24 home and office scenes. The stereo point clouds were found to be of good quality as they have been already stitched and aligned, reducing the need of preprocessing. All point clouds were denoised and downsampled to approximately twenty five percent of their original size. Voxel size was set to two centimeters; the maximum inlier distance for RANSAC was set to 1 cm, and clusters with a density less than 900 points were rejected to avoid false detection of noise.

4.2 Results and Discussion

The proposed system was able to achieve successful detection of chairs, sofas and tables as shown in sample results spanning in Fig. 4.1 to Fig. 4.3 and Tables 4.1 to 4.8.

The results are successful even in cases of partial occlusions, mainly due to the fact that ontological constraints can still be applied even if small sections of parts are occluded. There was no change in system parameters when experimenting through different datasets, with the exception of parameters related to clustering or down-sampling rates. This is due to non-uniform point cloud densities and cluttered objects, as well as to reduce required computational power.

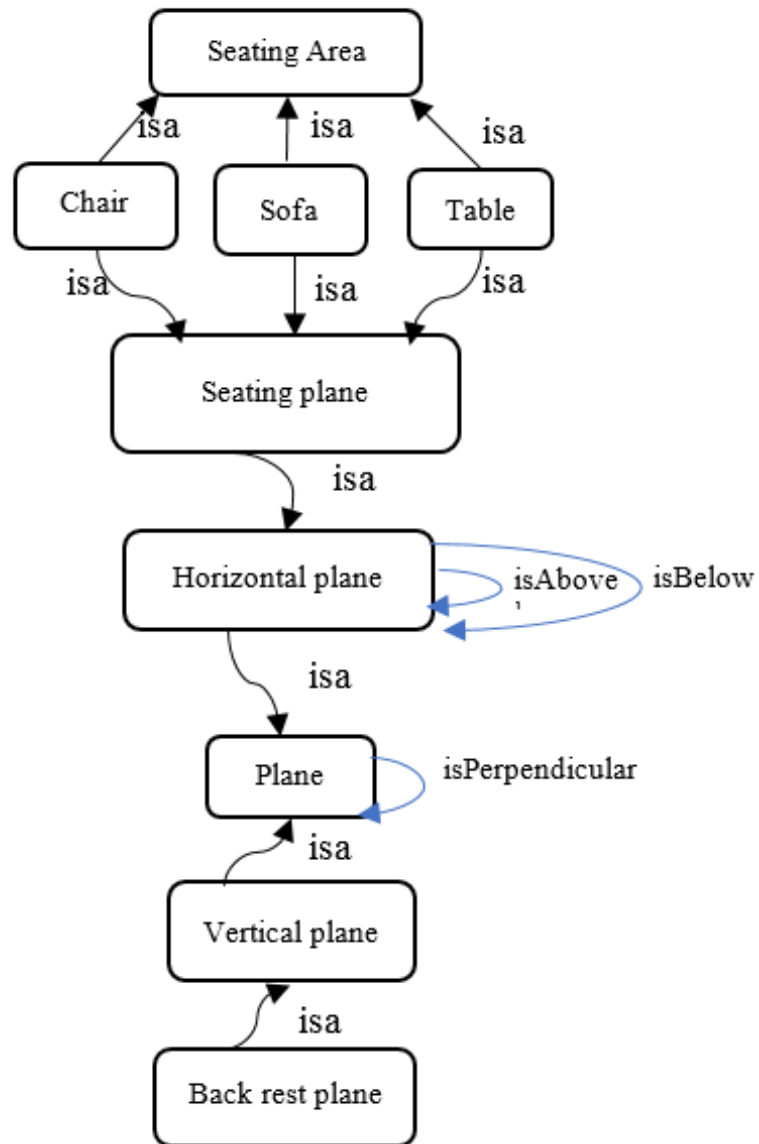


Figure 4.1: Sample ontology flow chart for a seating area. (Figure reproduced from [19])

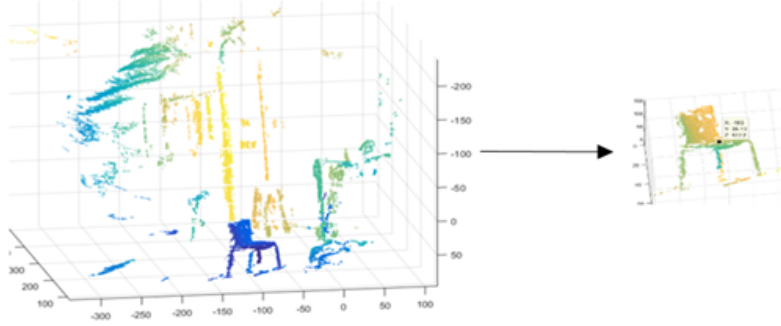


Figure 4.2: Scene from the vision and robotics lab showing a chair being detected by probabilistic ontology.

Table 4.1: System Parameters

Parameters	Unit	Min	Max
Plane Size	<i>cm</i>	25	150
Plane Inlier Distance	<i>cm</i>	0.5	1.5
Cluster Density	<i>points</i>	990	-
Height of Seating Plane	<i>cm</i>	35	75
Inclination Angle of Seating Plane	<i>degrees</i>	-6	6
Length of Cluster (in any direction)	<i>cm</i>	30	200
Downsampling Rate	-	20	25
Altitude cut-off	<i>cm</i>	-	140
Number of Planes in a Cluster	-	2	-
Threshold for Probabilistic Decision (%)	-	15	-
Neighborhood Radius for Density Based Clustering	<i>cm</i>	2	5
Voxel Size	<i>cm</i>	-	2

The training data for the SVM, used to estimate P_{in} and P_{out} consisted



Figure 4.3: Home scene from the dataset of Koppula et al. [1] showing successful detection of a sofa.

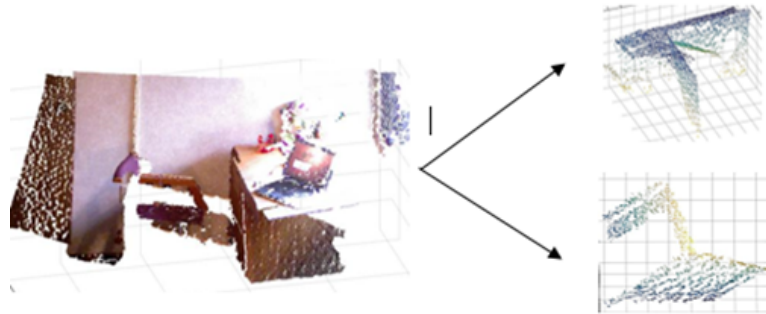


Figure 4.4: Office scene from the dataset of Koppula et al. [1] showing successful detection of a partially occluded chair and a false positive consisting of table desk.

of the MITs places205 [33] dataset, containing over 2 million images grouped into indoor and outdoor scenes. Combined with the GIST [34] descriptor, the trained classifier yielded a classification rate of 84.1%.

The general probability of having an object indoors or outdoors was evaluated using Stratified Sampling [35], whereas the values of $P_{obj/in}$ and $P_{obj/out}$ were successfully estimated by manual random query of indoor and outdoor images containing seating areas in the places205 dataset. After sampling of 235 indoor images and 201 outdoor images, a margin of error of five percent and a confidence level of 90% was achieved. A portion of the sampling data is shown in Figure 4.5 and Figure 4.6.

The ontology of a chair is so distinctive that our system produced very few false negatives. Tables, especially small ones are candidate objects that might be confused with chairs, however lacking a back support. After the effect of the optimization step, the proposed system has successfully assigned the label table. This is caused by the fact that most chairs and tables have four legs and both table and seating planes can sometimes be of near height however, as shown in Table 4.2, detected tables are often associated with lower probabilities than chairs,

given no or similar occlusions for both chair and table. The system is also found to be very sensitive to the quality of clustering. The authors have experimented with other clustering techniques such as region growing and supervoxels but could not yield better results than density based clustering. Clustering of point clouds remains an open research problem.

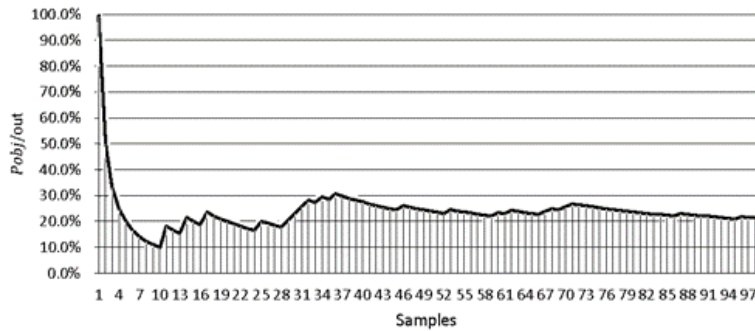


Figure 4.5: Statistical measure of the prevalence of seating areas in urban outdoors. The figure shows robust stabilization as of sample 78. The horizontal axis reflect the number of processed while the vertical axis shows the value of $P_{obj/out}$

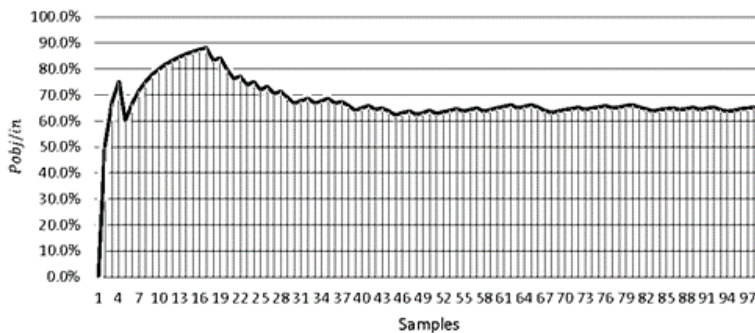


Figure 4.6: Statistical measure of the prevalence of seating areas found indoors in a random scene. The figure shows robust stabilization as of sample 49. The horizontal axis reflect the number of processed while the vertical axis shows the value of $P_{obj/in}$

Table II shows the generated cluster probabilities for one office scene from dataset 2. The falsely detected table has a lower probability than the detected chair, mainly because the seating plane of the table is higher and larger than common seating areas and it is lacking a back support. The cost of assigning the label l_3 =table turned out however to be the lowest of the three labels, as shown in table III. A comparison between single frame and full scene point clouds showed,

in contrast to precision rates, improved recall rates in full scene point clouds, and whereas single frame maps yielded improved precision rates. This is due to the fact that single frame scenes are more susceptible to partial exclusions at the boundaries, thus affecting the recall of objects, whereas full scene point clouds are more challenging to cluster. Precision data showed that our constraints are well balanced as shown in Tables 4.14, 4.15 and 4.16.

Table 4.2: Sample Cluster Probabilities of Fig. 4.4, Office Scene

<i>Reference</i>	<i>Ground Truth</i>	<i>Probability</i>
Cluster 1	Wall	$6.6 * 10^{-20}\%$
Cluster 2	Table	29.4%
Cluster 3	Chair	60.4%
Cluster 4	Closet	$1.2 * 10^{-18}\%$

The presented system outperformed all but the second dataset whereas precision slightly dropped below competition however, with better recall rates in office scenes. It is also noticeable that testing with home scenes yielded the worst results in both literature and current work. This is often due to closely packed objects in home scenes resulting in less accurate clustering. The addition of optimization improved precision by 4.5% on average, as some tables were identified and accordingly labeled. There were no improvements in recall rates after optimization.

Table 4.3: Comparison with Literature, Seminar Room Dataset

<i>Reference</i>	<i>Precision</i>	<i>Recall</i>
Gnther et al. [1]	81.0%	53.1%
Probabilistic Ontology	89.7%	69.2%

Table 4.4: Comparison with Literature, Office Scenes Dataset

<i>Reference</i>	<i>Precision</i>	<i>Recall</i>
Koppula et al. [8]	80.5%	72.6%
Probabilistic Ontology	71.4%	83.3%

Table 4.5: Comparison with Literature, Home Scenes Dataset

<i>Reference</i>	<i>Precision</i>	<i>Recall</i>
Koppula et al. [8]	57.8%	53.6%
Probabilistic Ontology	65.0%	61.1%

The below tables show improved results after energy function optimization. By using energy minimization for the decision making process, we are able to combine multiple factors into a single equation. Another advantage is the ability to add additional constraints to the problem: this is achieved by adding an additional term to the energy function. In this experiment, we used the latter advantage in order to discriminate between tables and chairs.

Table 4.6: Comparison with Literature, Seminar Room Dataset

<i>Reference</i>	<i>Precision</i>	<i>Recall</i>
Gnther et al. [1]	81.0%	53.1%
Probabilistic Ontology	89.9%	69.2%

Table 4.7: Comparison with Literature, Office Scenes Dataset

<i>Reference</i>	<i>Precision</i>	<i>Recall</i>
Koppula et al. [8]	80.5%	72.6%
Probabilistic Ontology	73.6%	83.3%

Table 4.8: Comparison with Literature, Home Scenes Dataset

<i>Reference</i>	<i>Precision</i>	<i>Recall</i>
Koppula et al. [8]	57.8%	53.6%
Probabilistic Ontology	70.1%	61.1%

Table 4.9: Cost of Assigning a label l to a Cluster Consisting of a Table

<i>Reference</i>	<i>Cost</i>
l_1 =Chair	1
l_2 =Sofa	1
l_3 =Table	0.7

Table 4.10 shows the results produced by manual sampling, in order to determine the likelihood of finding a seating area in a given environment. This done by randomly selecting over four-thousand images. For that purpose, Microsoft Excel has been used to input data and calculate the probabilities related to each environment. Figure 4.7 shows a sample of the generated curves using Excel, showing the probability evolution as we sample more images. A margin of error of five percent and a confidence level of 90% was achieved.

Tables 4.11, 4.12 and 4.13 show classification results for chosen scenes

from each dataset. This is done by running several images through 24 classifiers, then taking the mean score for each category.

Table 4.10: Probability of Finding a Seating Area

Variable	Reference	Available Data	$P_{obj/N}$
N_1	Airport Terminal	15100	27.0%
N_2	Auditorium	15100	64.7%
N_3	Cafeteria	5184	74.8%
N_4	Classroom	15100	55.5%
N_5	Conference Room	9154	89.7%
N_6	Corridor	15100	5.0%
N_7	Dining Room	15000	91.9%
N_8	Dorm Room	6202	15.5%
N_9	Food Court	7361	34.0%
N_{10}	Home Office	13942	55.4%
N_{11}	Hotel Room	15100	33.0%
N_{12}	Kindergarden Classroom	6395	20.5%
N_{13}	Kitchen	15100	19.6%
N_{14}	Kitchenette	15100	26.0%
N_{15}	Living Room	15100	98.7%
N_{16}	Lobby	15100	53.5%
N_{17}	Museum	7919	4.38%
N_{18}	Music Studio	15100	7.2%
N_{19}	Office	15120	58.0%
N_{20}	Reception	7311	16.7%
N_{21}	Shoe Shop	5184	4.8%
N_{22}	Staircase	15100	4.0%
N_{23}	Television Studio	5940	36.6%
N_{24}	Waiting Room 30	5921	91.7%

Table 4.11: Environment Classification P_N for Office Scenes

Variable	Reference	Available Data	P_N
N_1	Airport Terminal	15100	4.2%
N_2	Auditorium	15100	8.3%
N_3	Cafeteria	5184	8.4%
N_4	Classroom	15100	9.2%
N_5	Conference Room	9154	3.8%
N_6	Corridor	15100	3.1%
N_7	Dining Room	15000	10.7%
N_8	Dorm Room	6202	8.2%
N_9	Food Court	7361	6.4%
N_{10}	Home Office	13942	76.8%
N_{11}	Hotel Room	15100	1.5%
N_{12}	Kindergarden Classroom	6395	9.5%
N_{13}	Kitchen	15100	10.4%
N_{14}	Kitchenette	15100	8.7%
N_{15}	Living Room	15100	10.1%
N_{16}	Lobby	15100	9.4%
N_{17}	Museum	7919	3.3%
N_{18}	Music Studio	15100	7.1%
N_{19}	Office	15120	82.5%
N_{20}	Reception	7311	1.6%
N_{21}	Shoe Shop	5184	0.6%
N_{22}	Staircase	15100	0.0%
N_{23}	Television Studio	5940	4.8%
N_{24}	Waiting Room	5921	1.8%

Table 4.12: Environment Classification P_N for Home Scenes

Variable	Reference	Available Data	P_N
N_1	Airport Terminal	15100	6.7%
N_2	Auditorium	15100	1.9%
N_3	Cafeteria	5184	7.4%
N_4	Classroom	15100	4.9%
N_5	Conference Room	9154	6.4%
N_6	Corridor	15100	5.9%
N_7	Dining Room	15000	8.4%
N_8	Dorm Room	6202	6.6%
N_9	Food Court	7361	1.9%
N_{10}	Home Office	13942	6.7%
N_{11}	Hotel Room	15100	4.6%
N_{12}	Kindergarden Classroom	6395	1.7%
N_{13}	Kitchen	15100	5.9%
N_{14}	Kitchenette	15100	9.4%
N_{15}	Living Room	15100	92.1%
N_{16}	Lobby	15100	6.4%
N_{17}	Museum	7919	1.9%
N_{18}	Music Studio	15100	2.7%
N_{19}	Office	15120	6.4%
N_{20}	Reception	7311	2.6%
N_{21}	Shoe Shop	5184	0.9%
N_{22}	Staircase	15100	0.5%
N_{23}	Television Studio	5940	2.7%
N_{24}	Waiting Room	5921	6.7%

Table 4.13: Environment Classification P_N for Seminar Rooms

Variable	Reference	Available Data	P_N
N_1	Airport Terminal	15100	5.3%
N_2	Auditorium	15100	7.1%
N_3	Cafeteria	5184	6.4%
N_4	Classroom	15100	10.1%
N_5	Conference Room	9154	84.3%
N_6	Corridor	15100	2.7%
N_7	Dining Room	15000	2.9%
N_8	Dorm Room	6202	2.8%
N_9	Food Court	7361	3.7%
N_{10}	Home Office	13942	6.7%
N_{11}	Hotel Room	15100	4.9%
N_{12}	Kindergarden Classroom	6395	2.2%
N_{13}	Kitchen	15100	3.8%
N_{14}	Kitchenette	15100	1.9%
N_{15}	Living Room	15100	1.8%
N_{16}	Lobby	15100	8.6%
N_{17}	Museum	7919	2.7%
N_{18}	Music Studio	15100	1.8%
N_{19}	Office	15120	8.9%
N_{20}	Reception	7311	2.9%
N_{21}	Shoe Shop	5184	0.9%
N_{22}	Staircase	15100	0.0%
N_{23}	Television Studio	5940	2.8%
N_{24}	Waiting Room	5921	4.8%

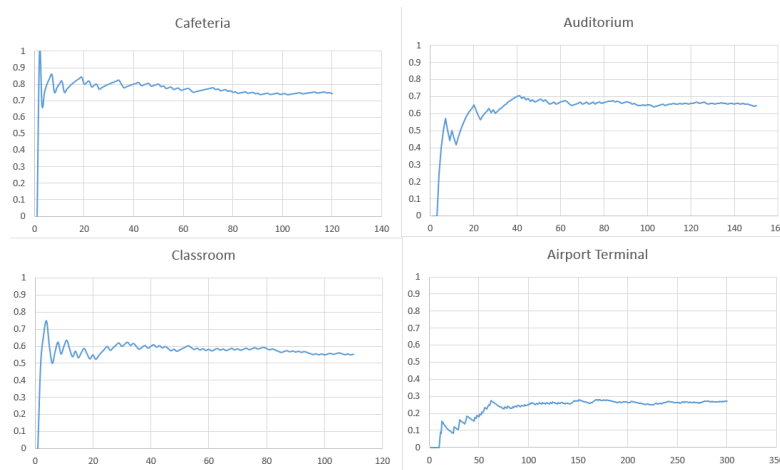


Figure 4.7: Sample Data Sampling for 4 out of 24 indoor categories. The horizontal axis represent the number of samples, whereas the vertical axis represent the probability of finding a seating place, on a scale from 0 to 1.

We now test our system for the third time using all three datasets, and we publish our results in the tables below:

Table 4.14: Comparison with Literature, Seminar Room Dataset

<i>Reference</i>	<i>Precision</i>	<i>Recall</i>
Gnther et al. [1]	81.0%	53.1%
Probabilistic Ontology	90.1%	68.2%

Table 4.15: Comparison with Literature, Office Scenes Dataset

<i>Reference</i>	<i>Precision</i>	<i>Recall</i>
Koppula et al. [8]	80.5%	72.6%
Probabilistic Ontology	79.1%	83.9%

Table 4.16: Comparison with Literature, Home Scenes Dataset

<i>Reference</i>	<i>Precision</i>	<i>Recall</i>
Koppula et al. [8]	57.8%	53.6%
Probabilistic Ontology	76.4%	62.1%

The advantage of knowing the environment substantially reduced false positives, mainly due to the fact that chairs found in unlikely environments held a higher cost for label assignment. Consequently, better values for precision were reached, in contrast to slightly varying recall rates.

For the purpose of generic CAD model matching, we chose a fourth dataset consisting of my own living room. Therefore, generic CAD models were roughly drawn using Solidworks then exported in STL format. Uniform mesh resampling was subsequently applied to the STL file using MeshLab, and exported as a point cloud in polygon file format.

The dataset was retrieved by capturing a 4 minutes video using a regular cell phones. Subsequently, over 850 frames were exported from the video. Using these images, the point cloud was generated using Structure from Motion "Visual SFM". The point cloud was subsequently manually scaled by comparison with ground truth information.

It has been found that generic object replacement is very sensitive to downsampling parameters. It is crucial to reduce the input point cloud inconsistency by using a grid based down-sampling method. For this experiment, a grid size of $0.5cm^3$ was used for both source point cloud and the meshed CAD models. We also found that CAD models will match better and quicker if they are slightly smaller than the detected objects. To prevent erroneous CAD model replacement, objects with very low probabilities are replaced with blank models, even if they are correctly labeled.

One unexpected challenge faced during matching is that surfaces detected by the capture device usually have a very thin thickness. This is due to the fact that the capturing sensors rarely has the opportunity to go into narrow places in order to capture the other side of some surfaces. In our example, this is true for most seating planes, since the camera did not travel (nor is it expected to do so) underneath the

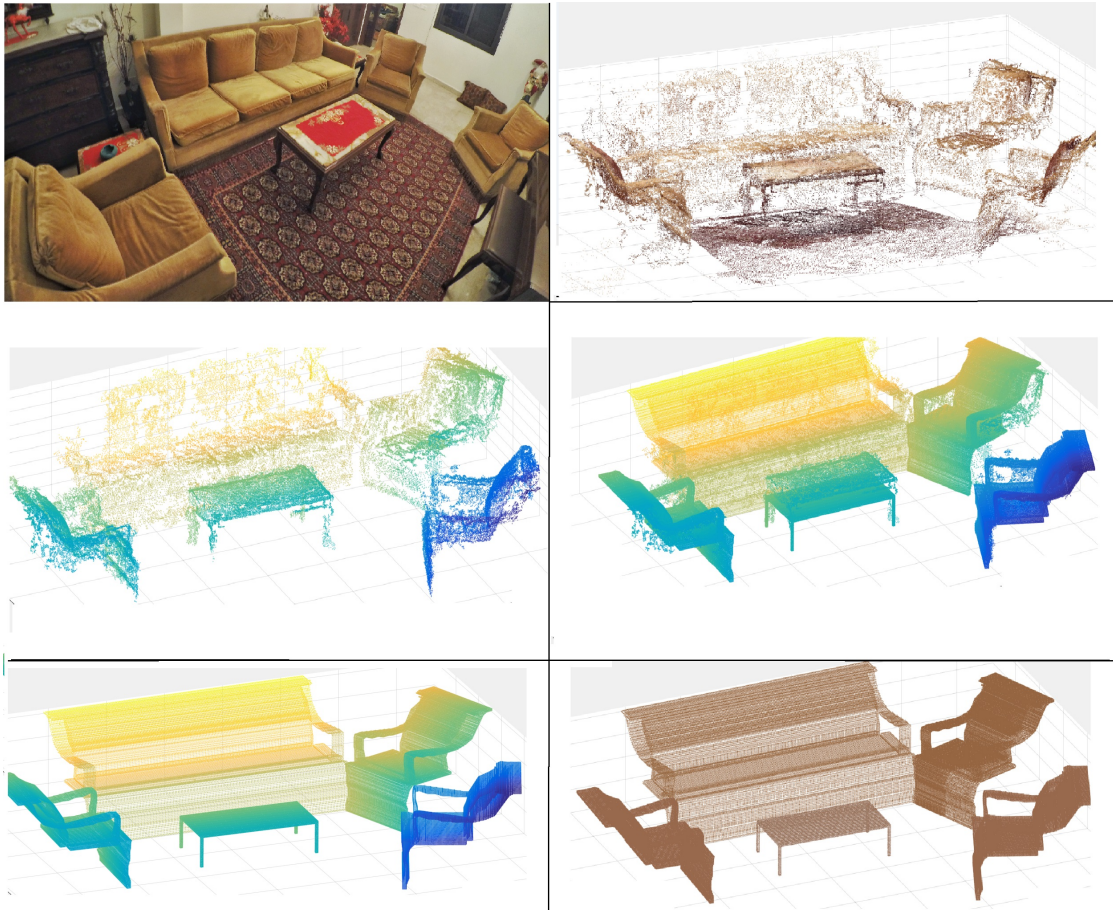


Figure 4.8: Left to right, the figure shows the original scene consisting of my own living room, followed by the captured point cloud using Visual SFM [2]. The second row features a map of detected objects by the same map, overlapped with CAD models. The bottom row features the final generic map, followed by the same map with color information restored according to the original point cloud colors.

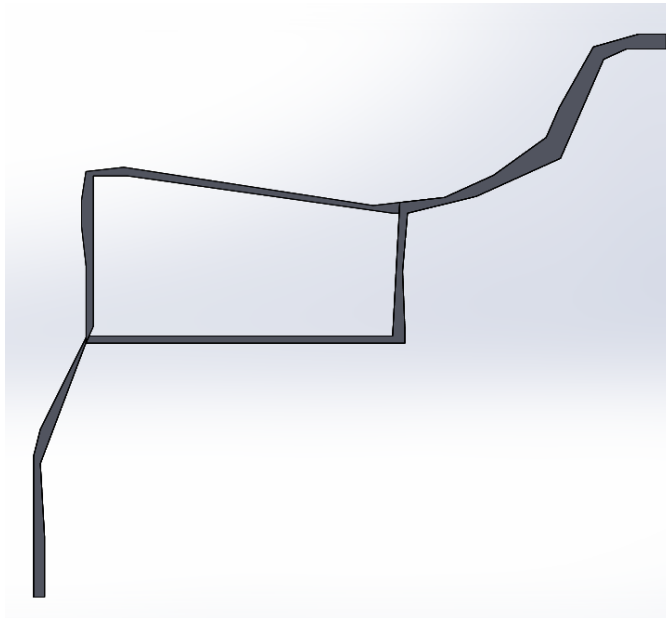


Figure 4.9: Skeleton of a chair, used to match with the point cloud of the detected chair.

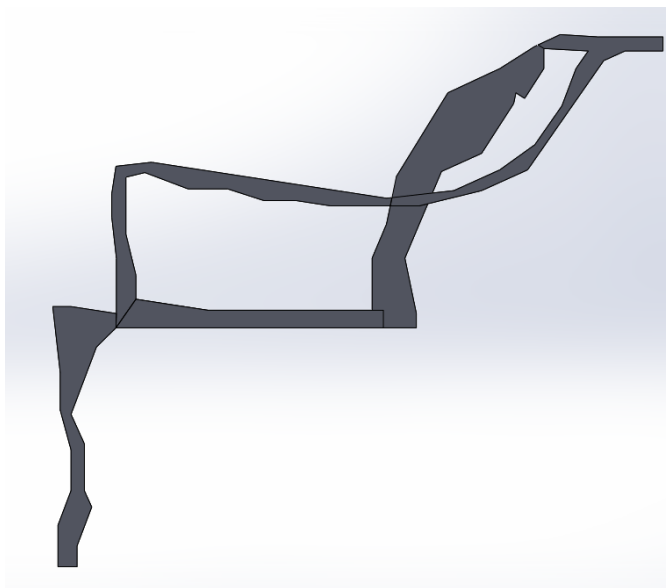


Figure 4.10: The transformation found by matching the skeleton to the object is then applied to the model in the figure above.

chair. We therefore obtain a thin profile for many surfaces. This challenge came to our attention in the experimental phase. To solve this problem, generic models were redrawn with very thin profiles. The modified models were then used for the purpose of finding the transformation matrix using ICP only. Subsequently, and since getting a map of thin objects is not as rewarding as getting a map of fully drawn objects, the transformation found earlier was thereafter applied to the generic objects drawn earlier, for visualization purposes. The figures below present examples of thin and non-thin profile generic CAD models. Finally, it has been found that constraining generic objects to the ground, and forcing both roll and tilt angles to zero will often improve the location and attitude of CAD models, even if they were not properly matched in their unconstrained versions.

Chapter 5

Conclusion

We have presented a novel framework for object based detection using probabilistic and ontological concepts, whereas an object is detected by constraining scene variables through ontological priors. Our method showed superiority in both precision and recall by comparison with relevant literature. Combining both ontology and probability holds promising ground for future work such as:

- Generate proposals for object class, rendering recognition a far easier task
- Build a map of semantic labels, then combining semantic information through a probabilistic ontological framework
- Build a map of generic objects that can be used for localization. Up so far, this work only addressed the perception problem through mapping of objects. An interesting contribution would include such a model in a Simultaneous Localization and Mapping (SLAM) algorithm. This is achieved through the use of transformed generic CAD models as landmarks for position correction.

Finally, lot of work can be done to transform decade old deterministic methods into better performing probabilistic models. This mostly due to the fact that Computer Vision is often associated with accounting for measurement errors and inaccuracies, which makes probabilistic approaches better fitting.

Appendix A

Abbreviations

SLAM	Simultaneous Localization and Mapping
RGBD	Red Green Blue and Depth
RANSAC	Random Consensus Algorithm
LSD SLAM	Large Scale Direct SLAM
URB	University Research Board
AUB	American University of Beirut
VRL	Vision and Robotics Lab
LVR	Las Vegas Reconstruction toolkit
CAD	Computer Aided Design
ICP	Iterative Closet Point Algorithm
IMU	Inertial Measurement Unit
PDF	Probability Density Function
RMS	Root-Mean-Square

Bibliography

- [1] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, “Semantic labeling of 3d point clouds for indoor scenes,” in *The Seventh Annual Conference on Neural Information Processing Systems (NIPS)*, 2011.
- [2] Y. Furukawa and J. Ponce, “Accurate, dense, and robust multi-view stereopsis,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.
- [3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.
- [4] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *International weekly journal of science*, pp. 436–444, 2015.
- [5] M. Gunther, T. Wiemanna, S. Albrechta, and J. Hertzberga, *Model-based furniture recognition for building semantic object maps*. Artificial Intelligence, 2015.
- [6] S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V. K. Papastathis, and M. G. Strintzis, “Knowledge-assisted semantic video object detection,” in *IEEE Transactions on Circuits and Systems for Video Technology*, 2005.
- [7] C. L. Griswold, “Platonic writings/platonic readings and penn state press,” *Penn State Press*, p. 237, 2001.
- [8] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena, “Contextually guided semantic labeling and search for 3d point clouds,” *IJRR*, 2012.
- [9] D. Purwar, A. Diwakar, and D. Bharti, “Image based real time object detection and recognition in image processing,”

- [10] M. Luessi, M. Eichmann, G. M. Schuster, and A. K. Katsaggelos, “Framework for efficient optimal multilevel image thresholding,” *Journal of Electronic Imaging*, vol. 18, no. 1, pp. 013004–013004–10, 2009.
- [11] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, pp. 679–698, June 1986.
- [12] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: a versatile and accurate monocular SLAM system,” *CoRR*, vol. abs/1502.00956, 2015.
- [13] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid, “Evaluation of gist descriptors for web-scale image search,” in *Proceedings of the ACM International Conference on Image and Video Retrieval, CIVR ’09*, (New York, NY, USA), pp. 19:1–19:8, ACM, 2009.
- [14] C. . I. S. Laboratory, “Regression & trend,” *National Center for Atmospheric Research*, 2016.
- [15] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, Inc., 1 ed., 1997.
- [16] S. Pillai and J. J. Leonard, “Monocular SLAM supported object recognition,” in *arXiv preprint arXiv:1506*, p. 01732, 2015.
- [17] A. Zouaq and R. Nkambou, “A survey of domain ontology engineering: Methods and tools.,” in *Advances in Intelligent Tutoring Systems* (R. Nkambou, J. Bourdeau, and R. Mizoguchi, eds.), vol. 308, pp. 103–119, 2010.
- [18] T. M. Bonanni, A. Pennisi, D. D. Bloisi, L. Iocchi, and D. Nardi, “Human-robot collaboration for semantic labeling of the environment,” in *Proceedings of the 3rd Workshop on Semantic Perception, Mapping and Exploration (SPME)*, pp. 1–6, 2013.
- [19] A. H. Selvatici and C. Anna, *Object-based Visual SLAM: How Object Identity Informs Geometry*. 2008.
- [20] J. Mason and B. Marthi, “An object-based semantic world model for long-term change detection and semantic querying,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura*, 2012.
- [21] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, “Slam++: Simultaneous localisation and mapping at the level of objects,” in *Computer Vision and Pattern Recognition (CVPR)*, 2013.

- [22] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “Rgb-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments,” *The International Journal of Robotics Research*, vol. 31, pp. 641–663, 2012.
- [23] G. Younes, D. Asmar, and E. Shamma, “A survey on non-filter-based monocular visual slam systems,” *arXiv:1607.00470*, 2016.
- [24] G. Nutzi, S. Weiss, and J. Scaramuzza, “Fusion of imu and vision for absolute scale estimation in monocular slam,” *Journal of Intelligent & Robotic Systems*, vol. 61, pp. 287–299, 2011.
- [25] P. Z. A. Torr, “Mlesac: A new robust estimator with application to estimating image geometry,” *Journal of Computer Vision and Image Understanding*, vol. 1, p. 138156, 2000.
- [26] M. Fischler and R. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [27] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley and Sons, 2006.
- [28] C. C. for Occupational Health and Safety, “Ergonomic chair-osh answers fact sheets,” *Government of Canada*, 2014.
- [29] J. Sander, X. Xu, E. Simoudis, J. Han, and U. M. Fayyad, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996.
- [30] T. Wiemann, H. Annuth, K. Lingemann, and J. Hertzberg, “An evaluation of open source surface reconstruction software for robotic applications,” in *16th International Conference on Advanced Robotics (ICAR)*, 2013.
- [31] T. Wiemann, K. Lingemann, A. Nchter, and J. Hertzberg, “A toolkit for automatic generation of polygonal maps-las vegas reconstruction,” in *European Conference on Mobile Robots, ECMR*, 2015.
- [32] A. Nchter, “6d slam - 3d mapping outdoor environments,” in *Journal of Field Robotics (JFR)*, pp. 669–722, 2007.
- [33] 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 27 (NIPS)*, 2014.

- [34] A. Olivia, “Modeling the shape of the scene: a holistic representation of the spatial envelope,” *International Journal of Computer Vision*, pp. 145–175, 2001.
- [35] R. M. Groves, *Survey Methodology*. Wiley, 2009.