SCENE ANALYSIS FOR INDOOR ROBOT NAVIGATION

By: AKLILEMARIAM RETA

A Thesis Submitted to the Department of Computing

School of Electrical Engineering and Computing

Presented in Partial Fulfillment for the Degree of Masters of Science in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

June 2020

Adama, Ethiopia

SCENE ANALYSIS FOR INDOOR ROBOT NAVIGATION

By: AKLILEMARIAM RETA

Name of Advisor: Prof. Yun Koo Chung

A Thesis Submitted to the Department of Computing

School of Electrical Engineering and Computing

Presented in Partial Fulfillment for the Degree of Masters of Science in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

June 2020

Adama, Ethiopia

Declaration

I hereby declare that this MSc thesis is my original work and has not been presented as a partial degree requirement for a degree in any other university, and that all sources of materials used for the thesis have been fully acknowledged.

Name: Aklilemariam Reta

Signature: _____

This thesis has been submitted for examination with my approval as a thesis advisor.

Name: Yun Koo Chung (PHD)

Signature: _____

Date of Submission: ___06/22/2020_____

Approval Page

We, the undersigned, members of the Board of Examiners of the final open defense by "*Aklilemariam Reta"* have read and evaluated his/her thesis entitled "*Scene Analysis for Indoor Robot Navigation*" and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirement of the Degree.

Name of Student	Signature	Date
Supervisor /Advisor	Signature	Date
Chairperson	Signature	Date
Internal Examiner	Signature	Date
External Examiner	Signature	Date
Head of Department	Signature	Date
School Dean	Signature	Date
Post graduate Dean	Signature	Date

Acknowledgment

I would like to thank the almighty God and blessed virgin Merry for the support I needed and for providing me everything I need in my life.

Next I would like to give my deepest appreciation to my adviser Professor Yun Koo Chung for his encouragement and support during the completion of this thesis s by giving me comments and suggestions. He provided me the motivation to work in the robotics and deep learning area.

I sincerely thank my family Amsale Asebe and Reta Alemu for providing me comfort and for being there when I needed them. I am also thankful for all my friends and my fellow Computer vision and robotics (CVR) SIG lab mates for the encouragement to finish my work.

Table of contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

LIDAR	Light Detection And Ranging
ROS	Robot Operating System
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SURF	Speeded Up Robust Features
ACF	Aggregate Channel Features

Abstract

Making mobile robots truly ubiquitous and cohabit with human beings require enhancing robot's ability to understand complex indoor environments. Service robots must perceive and understand complex indoor scenes and be able to recover room layout to better grasp the space and orientation of objects and 3D surfaces in the room. Robots ability to reason about the 3D surface have great implication for navigation and object detection. Robots must also take advantage of widely available information in indoor environment (i.e. text, sign door etc.) to solve long standing navigation problems like Loop closure problem (i.e. robot's inability to recognize a place it already visited. There has been very little work directed toward employing scene analysis algorithm for robot navigation and using text and sign to solve loop closure problem. So, introducing mechanism that integrate scene analysis and object detection algorithm will solve loop closure problem and improve performance of mobile robot navigation. This research proposes integrating scene analysis algorithm called room layout recovery and Aggregate Channels Features (ACF) object detector for mobile robot navigation. The algorithm first recover, classify, segment and label geometric surfaces (walls, ceiling and floors). Then the output of the algorithm is used to train Aggregate Channels Features (ACF) object detector. Which will be used to detect text, sign and doors. The algorithm is then implemented on turtle bot and simulated world to evaluate the effectiveness of the proposed algorithm. The proposed algorithm of integrating scene analysis and Aggregate Channels Features (ACF) achieves average precision of 0.7 & log average miss rate of 0.4 when door is partially visible and average precision of 1.0 and log average miss rate of 0.0 when door is fully visible. Additionally, the proposed algorithm achieves average precision of 0.9 and log average miss rate of 0.2 for text and sign detection. Moreover, the algorithm is tested using turtle bot in simulated world where it successfully detects door, text and sign. This research shows the importance of integrating scene analysis and object detection for robot navigation.

Keywords: *Indoor Object Detection, Scene Analysis, Scene Understanding, Aggregate Channel Features(ACF), Mobile Robot Navigation*

CHAPTER ONE

1. Introduction

1.1 Background

Robots are rapidly becoming integrated with people's daily life. Service Robots are deployed in ordinary people's homes and offices working and navigating in spaces shared with humans, perceiving the common surroundings, and provide support for elderly or disabled people. To accomplish many tasks robots must analyze and specify spatial arrangements and layouts in indoor scenes. Tasks such as search and indoor navigation require an understanding of space that is shared between robots and humans[1].

To understand the spatial orientation of the environment robots, build models of their surroundings through maps. Currently, Simultaneous Localization And Mapping (SLAM) is widely used for creating a map of the environment. SLAM enables the robot to concurrently create a map and locate the robot on the same map. Because the robot utilizes the same map to obtain its location, localization is error-prone[2].

Mapping algorithms could build reasonably detailed maps with the right hardware and conditions. But robotic mapping is usually tedious and existing methods for finding free space by recovering 3D geometry and exploiting the structure of indoor scenes turn out to be costly, laborious, and time-consuming[3]. Furthermore, because of error in measurement of sensors, partial visibility or occlusion of environment, similar and repeated patterns, limitation of range of sensors, and change in environment robots are unable to recognize the same place when they revisit that place again[2]. This problem is called a loop-closure problem, in which the robot is incapable of recognizing that it has already passed through the place earlier, and thus moving on a loop. For example, after a robot maps a room move to another place, and come back, it should detect a loop closure by comparing the landmarks[4]

One way to improve the efficiency of robot navigation and cohabitation with people at home and office is through increasing spatial layout awareness, 3D geometry, and structure of the indoor scene. To accomplish these incorporating scene analysis algorithms for robot navigation purposes would be relevant.

Scene analysis otherwise known as scene understanding aims at the interpretation of the content of an image. For example, consider the image of a typical living room with a sofa and chairs. After humans see the image they could not only identify the objects contained in the image like sofa chairs, floor, and walls but they could be able to estimate the entire space. If we equip our robots with the spatial understanding they would be able to reason about free space and improve their object reasoning[5].

To overcome the loop-closure problem other than exclusively depending on complex SLAM algorithm we could leverage abundantly available information's in indoor scenes like doors, text, and sign to uniquely identify and recognize places. Even though several algorithms are developed for particularly detecting doors, text, and sign respectively little has been done to use them together to improve navigation of robots in the indoor environment.

This research aims to utilize scene analysis algorithms to recover the spatial layout of rooms thereby increasing spatial awareness of robots and reasoning about free space. Moreover, incorporating text, sign, and door detection algorithm to identify the location of the indoor place which overall result would be the improvement of indoor navigation of robots. Additionally, I would like to investigate whether having prior knowledge of room layout would improve text, door, and sign detection, which in turn help solve the loop-closure problem.

1.2 Motivation

There are specialized algorithms developed for particular tasks in computer vision and robotics that focus on object detection, text detection, recovering 3d geometry, spatial layout, navigation, etc. One of my interests has been how can we combine and reuse algorithms to solve big problems they are not originally intended for and apply them in different contexts. how can different algorisms affect each other performance if they are trained on images that are already processed by other algorithms?

How could applications of computer vision improve the quality of people's life if mundane and repetitive tasks are carried out by service robots? furthermore, how cold algorithms developed for robotics can be used to solve problems faced by people with disabilities, elderly people, and people with visual impairment. Since service robots have great potential for helping people especially the disadvantaged, they should be aware of their surrounding space and utilize every piece of information available in indoor space including text, doors, and signs to fulfill their promise.

1.3 Statement of the Problems

Robot has to be able to move on their own and navigate their environment in order to fulfill the task they were given. Currently SLAM is widely used to plan, navigate and create map to model the space around them. Even though SLAM create a reasonably detailed map it has short coming when it comes to remembering the place it already visited, understanding room layout and incorporating abundantly information available in indoor environment like door text and sign to improve navigation of robots [2, 4].

Currently there a wide variety of literature that focus on recovering spatial layout of rooms, clutter and objects. Which is used to create 3d model recover free space from single image. Even though the importance for navigation is recognized there has not been wide effort to incorporate scene analysis for navigation[6].

. What differs indoor environment from outdoor environment is it relatively well-structured full of cues that identify the room including door, text, name plates and signs. Algorithms that specifically address the detection of doors text and sign are widely available. But utilizing these algorithms so that t the can be used to together to improve robot navigation specially identifying the place uniquely is not investigated [7],[8].

Recent works on scene analysis for indoor robot navigation have the following gaps

- Recovering room layout using Scene analysis algorithms is not widely applied for robot navigation [5, 9-11].
- The few work that exist only considers outdoor environment to apply scene analysis for robot navigation [6].
- Recovering free space from room layout [1, 3].
- Only focusing on floor segmentation and not including walls and ceiling to grasp the full room layout [12, 13].
- Loop closure problem: robot is unable to recognize a place it has already visited[2, 4].
- Text, sign and door are widely present in indoor environment. They can identify rooms but not explored to solve loop closure problem[7],[8].
- Only detecting doors not incorporating text and sign while most of the time they placed on name plates that are placed on doors [14].

- Few work exist for detecting texts and sign for blind person navigation but not applied for robot navigation [15].

In order to overcome these problems, it is important to incorporate scene analysis algorithm with robot navigation to recover room layouts and free space. More ever text, sign and door detection algorithms would enhance robot's ability to identify place thereby over coming loop closure problem.

In this study, the research attempt to answer the following research question.

- Doe having prior knowledge about spatial layout of room improve text, sign, and door detection algorithms?
- How to utilize text, sign and doors to identify the location of a robot in a room?

1.4 The Objective of the Study

1.4.1 General Objective

The general objective of thee research is to develop scene analysis algorithm that enhance spatial awareness of robots and incorporate sign, text and door detection algorithm to specifically identify and locate rooms for indoor robot navigation.

1.4.2 Specific Objective

The following specific objectives are addressed to achieve the general objective.

- Conducting literature review to understand the area and find out the gaps in works that are done by others researchers
- labeling image dataset for training and testing.
- Training room layout recovery algorithm
- Training aggregate channel features algorithm that can detect text, sign, and door
- Integrating room layout recovery algorithm with aggregate channel features algorithm.

- Evaluating the effect of the size of the training image on performance.

1.6 Scope and Limitation

1.6.1 Scope

The scope of the proposed work within the given time and resource includes

- Recovering the spatial layout of the room
- Detecting door, sign and text in indoor environment
- Integrating scene analysis with object detection (text, door and sign)
- Demonstrating the importance of incorporating scene analysis for robot navigation.

1.6.2 Limitations

This paper does not cover the following due to time and resource limitations

- Because of time constraint the research conducted in simulation
- Text detection is conducted only in English and does not incorporate local language because of low availability of data in local languages.
- The signs used in these research are limited to signs that show directions (i.e. arrows pointing diffident direction) and bath room signs due to time constraint.

1.7 Significance of the Study

The application of the research will significantly enhance robot's ability to reconstruct its surrounding which enable it to recover frees pace, avoid obstacle and navigate safely and autonomously. Robots will be able to segment and differentiate between floor, wall, celling and objects or obstacles placed on the floor. Which greatly enhance the navigation capability of the robot.

Robots will be able to cohabit and cooperate with humans easily, taking orders and fulfilling their duty by easily identifying and locating rooms in office or homes. This have great importance especially for service robots which require closely working with humans and operating indoors.

Robots will be able to easily navigate by putting names on doors, like kitchen, living room or the name of person in offices.

Service robots will be able to conduct home chores freeing people time to use it for more productive endeavors. Service robots can automat mundane and repetitive tasks This would have great impact for elderly people, people with disabilities, people with visual impairments.

Some of the application of this research

- The algorithm developed for this research could be utilized to improve navigation of people with visual impairments
- Service robot's will be able to assist elderly people with algorithms developed or replace people who take care of elderly people
- A message delivery robot in office environment could use the algorithm developed in these research
- Cleaning robots will benefit from these research
- The algorithm cold be applied for robot that serve as a wheelchair

1.8 Organization of the Thesis

The thesis is organized as follows:

Introduction (Chapter One) contains a little bit of the background of the research and the motivation for caring out the research. The statement of the problem and the research question are also included. The other topics are the general and specific objectives, the scope and limitations, and the significance of the study.

Literature Review (Chapter Two) gives an overview of scene analysis or understanding, object detection specifically text detection, sign detection and door detection algorithm. Different technique for mobile robot mapping, localization and navigation. It also covers the related work that closely try to solve similar problems as in research in more detail.

The methodology of the study (Chapter Three) explore the methods selected for this study and explain it in some detail. Data set, simulation, visualization, prototype and development tools are also discussed. furthermore, evaluation methods are also included.

Proposed Solution (chapter Four) presents the scene analysis algorithm in more detail and how it is integrated with detection algorism and mobile robot navigation.

Implementation (chapter Five) show how the algorithms are implemented, the setup of the working environment and prototype for scene analysis and mobile robot navigation. It also discusses how they are integrated with each other.

Result, Evaluation, and Discussion (chapter Six) present the result of integrating scene analysis with text, sign, and door detection and its importance for robot navigation.

Conclusion and Future Work (Chapter Seven) summarizes the work and conclude the result. It also presents future work.

CHAPTER TWO

2. Literature Review

2.1 Introduction

Computer vision is an interdisciplinary scientific field and a branch of artificial intelligence that deals with how computers gain insight and high level understanding from digital images and videos. To achieve these machines are trained to deduce and comprehend the visual world. by means of digital images from cameras, videos and internet computers are trained using traditional machine learning and deep learning algorithms, as result computers and robots will be able to accurately categorize and organize objects and then respond to what they perceive. These abilities have great importance for service robots.

Service robot are robot s which functions partially or fully independently on its own to implement valuable services to improve safety and quality of life of humans, they do not include industrial procedures, but they are able to accomplish pre-determined tasks. Example of personal service robots, include vacuum cleaning robots, lawn-mowing robots, elder care and medical companion robots, entertainment and leisure robots, including toy robots, hobby systems and kits, and home education and training robots are examples of personal service robots which are typically run by a untrained individuals

Currently there are multiple commercially available and successful service robots in the market. but to adapts service robots for complex indoor environments make them ubiquitous they should be able to build a model that allows them to analyze and understand surrounding environment of indoor scenes beyond object detection.

Vision is the most important sense for navigating and understanding the surrounding environment. For a service robot, the detection and recognition of objects are one of the significant operations it must accomplish. but object recognition alone don not give us the full picture. Therefore, analyzing the scene as whole to understand spatial orientation the room, localizing objects in scene, finding occlusion and free space in scene and recovering semantic relationship that exit between objects and the scene is necessary.

2.2 Scene analysis

Humans beings are tremendously skillful at visually identifying natural scenes and understanding the underlying scene structures from the image. To develop full understanding of a scene incorporating significant information from multiple levels by extracting semantic interactions, patterns, 3D spatial layout, objects and scene category is important. The relations between numerous objects and the scene they are located in is most intuitive and ordinary to human being. But it is very complicated for a computer vision system to replicate or mimic this task. More ever compared to object recognition, scene analysis tries to identify and locate the target objects and also the dispersal of targets in a scene [16].

In computer vision it is well known that images are numbers arranged in a grid to machines like robots or a computer. If we would like to enable our computers to have understanding of the content of an image like humans do, it will be necessary to extract the geometric and semantic clue embedded in the images, for example, given an image of indoor scene, a vision-based robot should be able to recover the complete 3D spatial layout of the room, semantic labels of the scene and its constituent objects. Furthermore, it is also required to understand the relationships present between different elements of the scene, scene and objects, effects of occlusion, scene and free space. These abilities are central to the way humans understand visual images, thus teaching artificial agents (AI) to have these abilities has been established goal in computer vision discipline [17].

Muzammal Naseer et al [17]formally define Scene Analysis (Understanding) as: **"**To analyze a scene by considering the geometric and semantic context of its contents and the intrinsic relationships between them."

The general aim of a scene analysis algorithm is to enable Artificial Intelligent (AL) agent's ability to understand a scene by detecting and categorizing scenes, localizing objects, recovering spatial layout and 3D geometry, extracting the semantic relationship to name some of them. Furthermore., other than detecting the visual features from images it is essential to build robust computer vision algorithms that can learn, adapt alternative explanations and approaches for analysis and interpretation visual scenes[18].

2.2.1 Approaches to solve scene analysis problem

There are different approaches that have been tried to solve scene analysis problem some of has be discussed below.

2.2.1.1 Scene classification

Scene classification is one of the most essential tasks for visual scene understanding. Insight about the scene or object category like outdoor or indoor, malls or office, forest or plain field would assistance in more complicated tasks such as scene segmentation and object detection. Classification algorithms are being used in diverse areas such as medical imaging, self-driving cars and context-aware devices. These approaches use different set of strategies including automatic feature learning, unsupervised learning and work on different 3D representations such as voxels and point clouds[17].

2.2.1.2 Object Detection

Algorithms that deal with recognizing occurrence of instances of object and their corresponding categories are called object detection algorithm. Often, these algorithms produce both the location, that is bounding box around the object and it respective class. For example, car, tree, television, dog, cat etc. This have huge importance for navigation of robots, blind person's navigation and self-driving cars. Nevertheless, to incorporate object detection in mobile robot navigation, we need known as 'amodal object detection' that not only attempts to discover the location of the object but also recover the objects whole shape and alignment in 3D space while only a portion of the object is visible. There a wide variety technique that are employed to solve object detection problem including handcrafted features, deep neural networks, region proposal, supervised models, unsupervised 3D object detection techniques[17, 19].

Developing object detection algorithm is a changing problem because of various reasons. For example, the actual world we try to model are full of clutters and object detection in such environments is very difficult. Moreover, in numerous cases, it is required to understand the scene setting to successfully detect objects. The effectiveness of an object detection algorithm can decrease as a result of discrepancies in object shapes, viewpoints, illumination, texture, and occlusion[19].

2.2.1.3 Semantic Segmentation

Semantic segmentation is the process by which we label each individual pixel with its semantically correct category. Pixel level grouping and categorizing necessitates knowledge about local and global therefore it would be necessary to build algorithms that can integrate the extensive contextual information together. The major difference between Semantic segmentation and object detection is that Semantic segmentation gives a pixel level classification in an image, that is it classifies the pixels into its corresponding classes, whereas object detection classifies the patches of an image into different object classes and create a bounding box around that object. Semantic segmentation could be applied in domestic robots, content-based retrieval, self-driving cars and medical imaging[20].

Conventionally, Conditional random fields (CRFs) used to be the most widely used algorithm for semantic segmentation. Mostly because that CRFs make available a flexible framework to model contextual information. But recently various forms of Convolutional neural networks (CNN) has been deployed instead of CRFs[17].

2.2.1.4 Physics-based Reasoning

Scene could be defined as a still image of the pictorial world or photograph. Nonetheless, when humans see at the still image, they can reason and deduce concealed dynamics in the scene. For example, consider a photograph of a football field with football players in a match, humans are able to recognize and guess the previous motion patterns and anticipate the upcoming events which are likely to occur in that a scene. As a result, people can plan for the next move and take knowledgeable decisions. By taking inspiration from this, research has been conducted in computer vision to build algorithms that can understand the underlying physical properties of a scene. These include guessing together the current and future dynamics from a still scene, understanding the support relationships and stability of objects, volumetric and occlusion reasoning[17] ,[21].

2.2.1.5 Object Pose Estimation

Object pose estimation tries to reconstruct the objects location in the image and its corresponding orientation with regard to a chosen coordinate system. Industrial robots and service robots that have requirements to operate on objects and perform different kind of manipulation on the objects need information about the objects location and its corresponding pose, which is very vital for

object manipulation by artificial agents(AI) and scene reconstruction. For example, by fitting 3D CAD models.it important to remember that object pose estimation task is closely related to the object detection task, consequently current research try to solve both problems successively or in a combined framework[17].One of the major methods employed to solve object pose estimation is called direct feature matching which tries find similarities between corresponding images and models[22].

2.2.1.6 3D Reconstruction

Ordinary people can see the world and understand the immediate environments in three dimensions (3D). Understanding the 3D spatial layout of a scene and the objects in it gives us a profound understanding of the mechanics, shape and 3D texture of objects[17]. Therefore, it is usually necessary to recover the complete 3D shape from images. 3D reconstruction is valuable in several areas. For example, robot navigation, medical imaging, computer graphics and virtual reality. Trying to reconstruct a scene from a singular or a set of RGB-D images with partial occlusions result in imperfect information. However,3D reconstruction from closely overlying RGB-D views of an object are comparatively an easier problem[23].

2.2.1.7 Saliency Prediction

The visual system of human beings exclusively focuses on salient parts of a scene and accomplishes a thorough understanding for the most salient areas of that scene. The discovery of salient parts of a scene corresponds to significant objects and actions in a scene and their joint interactions with each other. Saliency estimation and prediction methods and sensing modalities vary from research to research but most of them can be categorized into including RGB-D[24], stereopsis, light-field imaging and point clouds. Saliency prediction can be widely used in different areas including analyzing user experience, summarizing content of scene, automatically tagging image, automatically tagging video, privileged processing on resource constrained devices, object tracking and novelty detection. Predicting and estimating the saliency of images is a challenging task because, it is an intricate function of diverse features including appearance, texture of image, properties of background, location, depth etc. Which is complex to model these complicated relationships. Furthermore, saliency prediction needs both top-down and bottom-up clues to accurately model objects saliency. Therefore, the important requisite is to sufficiently model the local and global context[17, 24].

2.2.1.8 Affordance Prediction

Object detection and semantic segmentation algorithms could capture the relationship between objects more efficiently like the table is close to the shelf. Nonetheless, another fascinating way to understand indoor scenes is by considering the purpose or functionality of the objects which is called affordances of objects in technical terms. That means deducing what activities can be completed by utilizing a particular object. For example, people can sit on a sofa, people can put a tea mug on a table top. These particular features of objects can be utilized as characteristics, which have been found out to be important to transfer knowledge across categories[25]. Acquiring this kind of skills are essential for a wide variety of application. For example, they can be applied in areas including domestic service robots, industrial robotics, where the robots need to dynamically cooperate with the neighboring environments[17, 25].

Affordance prediction is a changeling task because, it would be necessary to gather information from numerous sources and reasoning about the content of the image to learn relationships. Moreover, it is important to model the hidden context to forecast the precise affordances of objects. Understanding the physical and material properties is central for affordance detection[17].

2.2.1.9 Recovering Spatial Layout

Scene analysis or understanding necessitates not only separately assessing individual elements of the visual domain but also considering the interaction between them. It will require building algorithms that can detect objects by considering underlying 3D geometric structure in the image. the spatial layout of 3D scene can be captured by building models about the interaction of objects, surface orientations, and camera viewpoint [10].

People can instinctively interpret and understand the 3D layout of surfaces and objects inside the scene. Peoples ability to understand the spatial orientation encompasses not only the visible surfaces but also include occlusion. For example, consider there is an image that shows table and chair in a living room. People can demarcate the chair and the table inside a room and perceive that the chair partially occludes the table .and that there is vacant space between them. Enabling computers with a comparable level of image based spatial understanding is one of the major challenges of computer vision[9]. The capability to guess the scene's 3D geometry and recover spatial layout is significant for a lot of tasks, including service and industrial robot navigation and object placement/manipulation, object detection and automatic single-view reconstruction[5].

Before recovering the spatial layout of the room we have to decide how to represent scene space. This is called parameterize the scene space. There are different techniques of parametrizations. Some of them can be categorized as follows according to[5]:

- a predefined set of prototype global scene geometries;
- a gist of a scene describing its spatial features;
- a 3D box or group of 3D polyhedral;
- boundaries between ground and walls;
- depth-ordered planes;
- constrained arrangements of curves;
- a pixel tagging of estimated local surface orientations, probably with arrangement constraints; or
- depth approximations at each pixel.

Room layout estimation and recovering 3D surface from single image has been an active topic of research over the past several years. Researchers employed different approaches to recover surface layouts. For example, fitting floor and wall borders in a viewpoint image taken by a flat camera to create a 3D model under Manhattan world expectations [26].The Manhattan world assumptions are that all walls are at right angles to each other and perpendicular to the floor. A special case of the Manhattan world is the cuboid model when four walls, ceiling, and floor enclose the room. Another method is producing Orientation Maps, generate layout hypotheses based on detected line segments, and select a best-fitting layout from among them[27]. Furthermore, recovering cuboid layouts by solving for three vanishing points, sampling layouts consistent with those vanishing points, and selecting the best layout based on edge and Geometric Context.[5] . Succeeding works follow a similar methodology, with enhancements to layout generation, features for scoring layouts, and incorporation of object hypotheses or other context[28].

In recent times, there has been much progress in getting better understanding about 3D models of a scene from single image and perceiving the spatial layout of rooms and the objects located inside the scene. Researchers have prepared numerous models that able to recover scene properties. For example, primal sketch, surface orientations, depth illumination and occlusion boundaries. The scene analysis framework described by Hoiem et al [29] permits numerous diverse visual tasks to work collectively by iteratively connecting them through their intrinsic images. Their [29]

experiments demonstrate that the approach they utilized is effective, it shows that inference over graphical models may not be obligatory. With appropriate selections of learning algorithms in each of the respective task the framework [29] have the additional benefit of sharing features between algorithms. For instance, a separate algorithm could take advantage from the structural knowledge of the input space that is learned by different algorithm.

2.2.1.10 Holistic or Hybrid Approaches

In this review so far as explained above there are individual tasks that are significant to solve a scene analysis problem and advance understanding of indoor scenes like semantics segmentation of scene, objects detection and localizations, learning about object functionalities and saliency of scene. Hybrid or holistic scene understanding algorithms try to build a model that can concurrently reason about many complimentary characteristics of a scene and provide a thorough scene understanding capability. To be able to build systems that can incorporate separate tasks into integrated system can lead to concrete systems that can operate in real-life scenarios such as robotic platforms interacting with the real world. For example, computerized systems for danger detection and quick response and rescue[30].

Each individual tasks that described above are in fact are not separate but interdependent on each other. Consider for instance, prior knowledge that an image is an indoor scene, could be used to improve the estimate of the depth from a single image more precisely. Additionally, if consider a service robot or any other robot whose duty include grasping objects, information about the nature of the object that the robot is trying to grasp, would make it easier for the robot to come up with solutions on how to pick it up and manipulate it[31].

In [31] they developed a framework for scene understanding called Cascaded Classification Models (CCM) treating each classifier as a black-box. Each classifier is recurrently instantiated with the subsequent layer with the outputs of the preceding classifiers as inputs. [31]While this work proposed a method of joining the classifiers in a way that amplified the performance in all the tasks they considered, [31]it had a drawback that it optimized for each task independently and there was no way of giving back information from later classifiers to earlier classifiers during training. This feedback can potentially help the CCM achieve a more ideal solution.

In[32] they proposed a Feedback Enabled Cascaded Classification Model (FE-CCM), which combines different classifiers trained for an explicit task. For example, object detection, event

detection, scene classification and saliency prediction. This combination is accomplished in a cascaded manner with a feedback mechanism to cooperatively learn different task by specific models for the purpose of scene understanding and robot grasping. They [32] argued that with the feedback mechanism, FE-CCM acquires knowledge about significant relationships between sub-tasks. An essential advantage of FE-CCM [32] is that it can be trained on diverse datasets meaning it does not require data points to have labels for all the tasks.

The challenges for building holistic scene analysis algorithms are the following. First precisely demonstrating the relationships between objects and background is a tough task in real-life environments. Moreover, effective training and interpretation is problematic because of the Combination of several individual tasks and supplementing single source of information with additional information is a crucial challenge[17].

2.3 Scene analysis for robot navigation

A considerable number of research has been conducted during the previous years in pursuit of solving obstacle detection, avoidance and ground plane segmentation problem which employ different scene analysis algorithms.

On the topic of algorithms that utilize monocular or single camera there are a range of techniques that were proposed by researchers including, applying a mixture of color and gradient histograms to differentiate free space from obstacles, using a region-based obstacle detection technique for indoor navigation by using a lone color camera and delivering a local obstacle map at high resolution. More over applying adaptive framework that could learn the appearance of the ground during operation in real time. Color appearance is also employed to categorize individually different pixels as belonging to either to an obstacle or to the ground. Additionally, there are researcher's that use holography to estimate the ground plane normal after that the floor is detected by calculating the plane normal from motion fields[33].

In [34] they try to map appearance features straight into actions, like it is used to be applied in imitation learning and is currently being applied with end-to-end deep learning. This kind of methodology is certainly preferable if the training environment is analogous to the test environment.[34]demonstrate a remarkable example of this method, using a simulator of office-like indoor environments with powerfully changing textures. which perform very well when

applied in a real office environment. Nevertheless, utilizing the trained network to a totally dissimilar surrounding could result in a lesser performance.

In [35]they introduce a method that relates image appearance to estimate distances. They [35] applied a data set acquired through a laser scanner. Then by aligning it with the camera pose it is possible to study a mapping from feature vectors to a distance per pixel[35]. The most important improvements by this approach are gained by using deep learning algorithms. It is apparent that learning distance and depths from a training data set is not assurance that the algorithm performs better in the real test environment. For example, the absolute scale of the distance approximations could be erroneous by large margins if the training set had outdoor images whereas the test set contains of indoor images.

One of the earliest methodologies for obstacle detection is to assume the occurrence of a local ground plane. By establishing that a ground plane already exited and it is appropriately segmented in the image, it is possible to determine how far and which way the obstacle is located by considering the image coordinates at where the obstacles touch the plane. To apply this approach in a new and unidentified environment the algorithm has to be able to learn and adapt to the new environment. Initially the algorithm learns about the appearance of the floor by predicting that the floor directly in front of the robot does not contain obstacles. But [36]disputed this assertion by showing that this is a strong assumption, and demonstrated it is vital to also learn floor appearance further away, as the ground plane usually has dissimilar appearance higher up in the image. They [36] resolved this difficulty by using the fact that driving over the ground plane successfully was an evidence that it was free of obstacles. Therefore, the robot could consult its wheel odometry to recover the appropriate regions from previous images learning as a result floor appearance far away. The disadvantage of this methodology is that it will inhibit the robot from driving on a ground plane which has a different color from the already experienced one, limiting the robot ‘s movements

Additionally, [37]developed a methodology that segments the picture in to two separate classes of sky and non-sky. The idea depends on the way that non-sky areas over the skyline line are obstacles to flying robots This thought was first illustrated in[37]. The algorithm learned a decision tree based on a wide scope of computer vision features for grouping pixels as sky or non-sky.

Segmenting a particular digital images of natural scene is into semantically significant units' is established study problems in computer vision. Enabling the robot to differentiate and categorize surface in indoor scene like floor, wall celling is the foundation for solving a diversity of problems in robotics and navigation for example navigation, free space estimation, image enhancement, 3D reconstruction, scene understanding and classification[38].

Researchers have applied varying approaches and encountered different level of success in semantically segmenting and analyzing indoor scenes for robot navigation. For example, researcher's use innovative sensing devices such as Kinect and, standardized multi-image setups to excerpt depth data [39].Some researchers impose constraints on the environment like horizontal camera motion must be parallel to the floor without spinning or Manhattan world assumptions [39].

A minority of researchers have deliberated on the floor detection problem for robot navigation. Most of the methods reviewed utilize the ground plane constraint. [38]Some of the approaches implemented by researchers are planar homographs to optical flow vectors including stereo homographs, calculating simply a sparse representation of the ground plane by categorizing sparse feature points and applying a pixel wise choice to create a condensed ground plane representation[38].

Uncovering the floor from a particular indoor image is a challenging task due to the following problems[39].

- Clutter like objects are present
- floors and walls are textured
- shadow and highlights are present due to illumination or
- when clutter is not confined to wall-floor boundaries.
- Another challenge is scenes with non-Manhattan layout where adjacent walls are not perpendicular to each other.

2.4 Text and sign recognition

Object detection and recognition is an essential element for scene understanding. Robust and effective indoor object detection support robots to self-sufficiently navigate previously unknown indoor environments and evade hazards. To enable robot, perceive the environment like human beings, robots have to be able to processes and distinguish information like office and living room,

shutter, sign, text or door etc. This problem could be resolved by adding semantic information to a map created by the robot, which enhances navigation ability of the robot. additionally, it has benefit of realizing a improved human robot interaction[40].

Sign and text are predominantly better at discriminating among similar objects in indoor environments such as elevators, restrooms, exits, and office doors. only few academics have researched on applications of text and indoor sign for robotic navigation. As shown by [41] the approaches can be categorized as follows : edge-based text detection algorithm appropriate to office or home surroundings, detecting a set of identified landmarks where some include signs for example apartment figures, and modeling of the surroundings halls and doors so that the an artificially intelligent(AI) agent will be able to explore for a specified room by primarily matching the door model and then utilizing that to locate the target text. Additionally [41] describe a robot that is able to extract and identify text from signs of known font, size, and background in a controlled lab atmosphere without including mapping.

Early examination has utilized texture segmentation with heuristics to recognize text in non-document settings. Other work has applied machine learning where a boosted classifier is prepared to recognize text in road pictures. conventional object detection and item identification strategies for example, deformable parts models are likewise applicable to "word spotting" — discovering words from

Yang et al [42]proposed a context-based totally indoor item detection system as a resource to blind human beings for getting access to unknown surroundings. Their system consisted of a door detector and a text extractor algorithm for studying textual content on notice boards. The text extraction method used optical character recognition for studying the textual content displayed on indoor signs, name plates and notice boards. However not all indoor signs have textual content on them.

Yingli Tian et al [43] proposed a technique that contains both detection and recognition procedures. Detection procedure acquires the position of sign or text in the picture. Recognition procedure is then executed to recognize the detected sign or text. The sign and text detection is founded on effective shape segmentation. The sign and text recognition employs SIFT feature-based matching, which is robust to variations of scale, translation and rotation .They [43] join the window-sliding procedure with the classifier to detect regions of a picture at all areas and scales

that contain the given objects. Nonetheless, the window sliding technique experiences two weaknesses: 1) high handling time; and 2) incorrectness of detection results because of various background[43].

Indoor object detection one of the challenging tasks in computer vision domain because of the following issues[42]:

- there are great intra class variations of appearance and layout of objects in dissimilar architectural environments;
- there are moderately little between class varieties of various object models. the essential shapes of signs on a restroom, a leave, a research center, and a lift are fundamentally the same. It is hard to recognize them without utilizing the related context data;
- contrasted with objects with advanced texture and color in normal scene or outside situations, most indoor items are man-made with less texture. Existing feature descriptors which function very well for open air conditions may not successfully characterize indoor objects; and
- when the robot travels around the room, the progressions of position and separation between the user and the object will cause huge view varieties of the object, more over just piece of the object is caught. Indoor wayfinding help ought to have the option to deal with object occlusion and view varieties

2.5 Door detection

Indoor buildings like home and office are intended to be profoundly organized structures, in which the different parts (e.g., dividers, floors, entryways, doors and passages) are set in highly predictable way according to each other. Recognizing these semantically significant segments is critical for versatile mobile robot navigation,

There are a variety of approaches that employ door detection algorithm for mobile robot navigation. A portion of these strategies use laser range finders to set up sensor models of the general environment and to get the distance information to test door concavity[44]. others utilized sonar information to affirm or dismiss detection results from cameras.

In[45], three cameras are utilized to perform stereo vision for entryway discovery. In any case, expensive cost, high power utilization, and complexity in these frameworks make them unsuitable

to work for robot navigation. To decrease the expense and complexity of the device and its computational necessities using a single camera is preferable.

In [46]and [47]doors are recognized utilizing both visual data and range information (sonar). In [46] creators manipulate the way that vision is useful for giving long range data beyond the scope of ultrasound sensor and detect and cluster vertical lines dependent on the expected dimensions of the door and make preliminary door hypotheses.

In [47] the researchers tackle increasingly broad issue of acquiring a model of the environment characterized by instances of numerous objects of predefined class (for example doors, floors) given range information and color pictures from an Omni-directional camera. The doors are then detected as specific instances of the door model, given all the sensory information. The door hypotheses are acquired by fitting straight sections to laser range information and related color values from the omnidirectional camera

In [48] both laser information and cameras were coordinated in such way that trinocular vision framework was utilized to choose a potential door initial location and laser estimations are permitted to progressively refresh the door location while exploring towards the door.

In [49]researcher's concentrate around dealing with the varieties in door appearance because of camera pose, by characterizing properties of the individual segments utilizing semantic factors of size, direction and height and join the evidence utilizing fuzzy logic.

Additional research utilizing visual data is accounted for in[50], where just geometric data about configurations of line segments is utilized. In many examples, just the doors which were plainly obvious and near the observer were selected as correct hypotheses.

There are a couple of existing door recognition algorithms utilizing monocular visual data[51-53]. In [51] an Ada Boost classifier is prepared to distinguish door of similar appearance by joining the features of sets of vertical lines, concavity, hole between the door and floor, color, texture and vanishing point. Be that as it may, a distinguishable hole beneath the door and floor isn't generally present in various situations.

Munoz-Salinas et al [52] built up a door frame model-based door detector by utilizing Hough Transform to extract the edge segments and a fuzzy framework to analyze the connection between the segments. In any case, their algorithm can't separate doors from huge rectangular objects

normally found in indoor environment, for example, shelves, cupboards, and pantries. In[53]. two classifiers were trained by utilizing color and shape features. This[53].algorithm was intended to recognize the doors of the office, where all the doors have similar color. It would fail if the color of the doors changes.

To overcome the limitations described above,[54] developed an image-based door detection algorithm by establishing a general geometric door model that utilizes the general and stable features of doors (i.e. edges and corners) is important. Furthermore, integrated with geometric information of lateral at similar horizontal coordinate, the proposed algorithm is able to distinguish doors from other objects with door-like shape and size. The detection results demonstrate that our door detection method is generic and robust to different environments with variations of color, texture, occlusions, illumination, scales, and viewpoints. But the method has not been applied for robot navigation purposes. These approach have difficulty detecting doors if the whole structure of door is not captured by the camera. These happens when the robot close to the door and couldn't capture the whole image.

A minority of work on monocular obstacle detection has focused on using visual appearance cues. One approach is to assume the presence of a local ground plane. If such a plane is present and well-segmented in the image, the distances and directions to ground obstacles can be determined by using the image coordinates at which the obstacles touch the plane. In toy-like environments, as in the robo cup competitions[55], a color can be pre-selected by a human, leading to a computationally efficient and accurate segmentation. However, if one wants to use this approach in a priori unknown environments, a learning component is essential.[56] learned the appearance of the floor by assuming that the floor directly in front of the robot was free of obstacles.[57] argued that this was a strong assumption, and showed that it was important to also learn floor appearance further away, as the floor often has quite a different appearance higher up in the image. They [57] solved this problem in an elegant way, by using the fact that successfully driving over the floor was a proof that it was free of obstacles. Hence, the robot could use its wheel odometry to retrieve the relevant regions from past images - learning also floor appearance further away. A drawback of this approach is that it will prevent the robot to drive on a floor of a different color than experienced before, limiting the robot 's movements

2.6 Mobile Robot Navigation

For mobile robots, the capacity to explore the surrounding environment is significant. The robot ought to maintain a strategic distance from circumstances like collusion, dangerous conditions, yet also should achieve its purpose to navigate in the surrounding robot environment. Robot navigation the capacity of a robot to arrive at an ideal area with the capacity to position itself in the current environment and plan a path to the desired location.

Mobile robots should be able to realize its own location and orientation within the frame of reference or coordinate and plan their path. Robot should be able to understand the current location and destination of the robot and planning how to navigate to the destination from the current location within the same frame of reference or coordinate. Moreover, robots must be able to build a Map of the environment. A map is the representation of the robot environment, in which the robot can refer to understand the environment layouts and location.

2.6.1 Simultaneous Localization and Mapping (SLAM)

SLAM deals with the problem of mobile robot navigation or building a map of an unknown environment while at the same time navigating the environment using the map and localizing itself[58]. It's where a versatile robot constructs its own guide and understands its present location all the while exploring the unknown environment. SLAM consists of different parts like Landmark extraction, data association, state estimation, state and landmark update. SLAM has many different steps and these steps can be implemented using a number of different algorithms[58].

2.6.1.1 Landmark Extraction

Landmark is a feature element which is utilized by the robot to discover where it is which is handily recognized from the environment[59]. A landmark must be detected or observable from different point of view and it must be unique for instance it should be easily distinguished from other landmark

2.6.1.2 Spikes land mark extraction

These kind of land mark extraction used for landscape changing between two laser beam or used for extracting landmark for non-smooth surface by using extreme value in which from two laser scan reading differ by certain values[60].

2.6.1.3 RANSAC land mark extraction

RANSAC land mark extraction used for smooth surface this types of land mark extraction used for extracting line from the laser scan by using least square approximation. After collecting the laser reading point RANSAC checks how many laser readings lie close to this best fit line if the number is greater than some thrash hold value which is called consensus it can assume that it finally gets line [59].

2.6.1.4 EKF (extended kalman filter)

In this kind of land mark extraction is mainly used for smooth surface this types of land mark extraction used for extracting line from the laser scan by using least square approximation[61] .After collecting the laser reading point RANSAC checks how many laser readings lie close to this best fit line if the number is greater than some thrash hold value which is called consensus it can assume that it finally gets line[62].

The first step in extended kalman filter is prediction step in this step it updates the current position using the odometry data. In the second step since the predicted current position by odometry is not exactly correct additional process is needed to compensate for these errors. This is done using landmarks. In this process try to predict where the landmark is using the current estimated robot position and the saved landmark position. After calculating error matrix Range should also be updated to reflect the range and bearing in the current measurements to compute the Kalman gain. Finally, it computes a new state vector using the Kalman gain[63].

2.6.1.5 Data association

In this process after the landmark extraction it will be stored in database. After it stores the extraction point, the robot starts for scanning other land mark using the laser scan and the data association start to associate the newly detect reading from already stored data in the database[64]. If the newly read data is new, then it will be stored as newly observed landmark in database to make it ready for the next association. By setting a constant, an observed landmark is associated to a landmark if the following formula holds [65].

CHAPTER THREE

3. The Methodology of the Study

3.1 Overview

In this chapter, the research methodology and the procedure to develop the scene analysis algorithm for indoor robot navigation is explained. Appropriate simulation, prototype and development tools will be described.

In order to accomplish the objective of this study, the subsequent methods and techniques are utilized.

3.2 Experiment setup

The experiment is conducted in simulation. The simulation is carried out on Linux platform using Ubuntu 16.04.2.1.A building containing multiple rooms is created for the experiment. Each room separated by doors and hallways. Furthermore, signs and texts are placed on the doors and hallways. The building is created using Gazebo software, which enables the users to create models that are used for simulation. Additionally, Rviz software is used to create map of the building and to simulate the navigation of the turtlebot inside the building.

3.3 Data set

The dataset contains images that represent indoor environment and the surrounding objects. To achieve the objective of the study images of doors, text and sign commonly found indoor environment are included.

The images of doors, text and sign is first collected from google image search. The size of the images is then set to 250 x 250 pixel and labeled using GIMP (GNU Image Manipulation Program).

To properly simulate indoor environment, additional to images that contain only sign, text and door incorporating the overall surrounding including hallways, floors, ceiling and walls is necessary. To achieve that images of corridors and hallways are taken from a dataset developed by Michigan university. The Michigan Indoor Corridor Dataset[66] is made explicitly to improve indoor navigation. It contains images of halls and workplaces in the college taken by a portable robot. The dataset contains images obtained with camera mounted on a mobile robot. The camera

was set-up so that there was zero tilt and roll angle with respect to the ground. The camera has a fixed height of0.47 m with the ground all through the video. The camera utilized was: AVT Manta G-145 Color CCD Camera 5mm FL Wide Angle Low Distortion Lens [66].The frames of the video are changed to images and put in zip file by the researchers which are used for training purpose.

The Michigan Indoor Corridor Dataset Contains 1,885 images in total used for training and testing. The sign data set contains 253 images for training 150 for testing. Furthermore, the text data set contains 247 for training and 150 for testing.

3.4 Scene analysis algorithm selection

To improve navigation capability of robots, recovering the surface layout and determining drivable region is of paramount importance. Because it enables the robot to distinguish the floor from the walls, avoid obstacles and separate free space. More over differentiating the walls from the floor and ceiling enable the robot to know where to look, when searching for doors, sign and text. As a result, scene analysis algorithms that focus on recovering spatial layout of the scene are better suited for these research.

For selecting the appropriate scene analysis algorithm for the proposed research, I am considering algorithms not only recovering the 3D surface layouts but also locate objects inside the scene. because clutter and obstacle are widely present inside rooms. Additionally, accuracy, specially error per pixel of the algorithm is considered.

There are mainly two approaches for recovering room layout. This are called geometric context (GC)and orientation maps (OM)[67].

- **Orientation Map**

  Albeit single pictures are a reliable information for indoor space modeling, automatic recognition of various structures from a single picture is extremely testing. Lee et al[68], introduced the orientation map for assessment of their produced layout hypotheses. The primary idea of the orientation map is to characterize which regions in an image have a similar orientation. An orientation of a region is dictated by the direction of the normal of that region. In the event that a region belongs with the XY surface, at that point its orientation is Z)[67].

- **Geometric Context**

  Hoiem et al, [69]labeled a picture of an open air scene into coarse geometric classes which is helpful for many activities, for example, navigation, object recognition, and general scene understanding. Normally the camera pivot is generally lined up with the ground plane, empowering them to reconcile material with perspective. They ordered each area in an outside picture into one of three primary classes. To start with, surfaces which are generally corresponding to the ground and can possibly support another solid surface. Second, strong surfaces those are too steep to even consider supporting an object. Third, all picture areas which are relating to the open air and clouds[67].

3.4.1Comparison of orientation map and geometric context

The following image show the comparison of accuracy between orientation map and geometric context by changing the horizontal view angle.

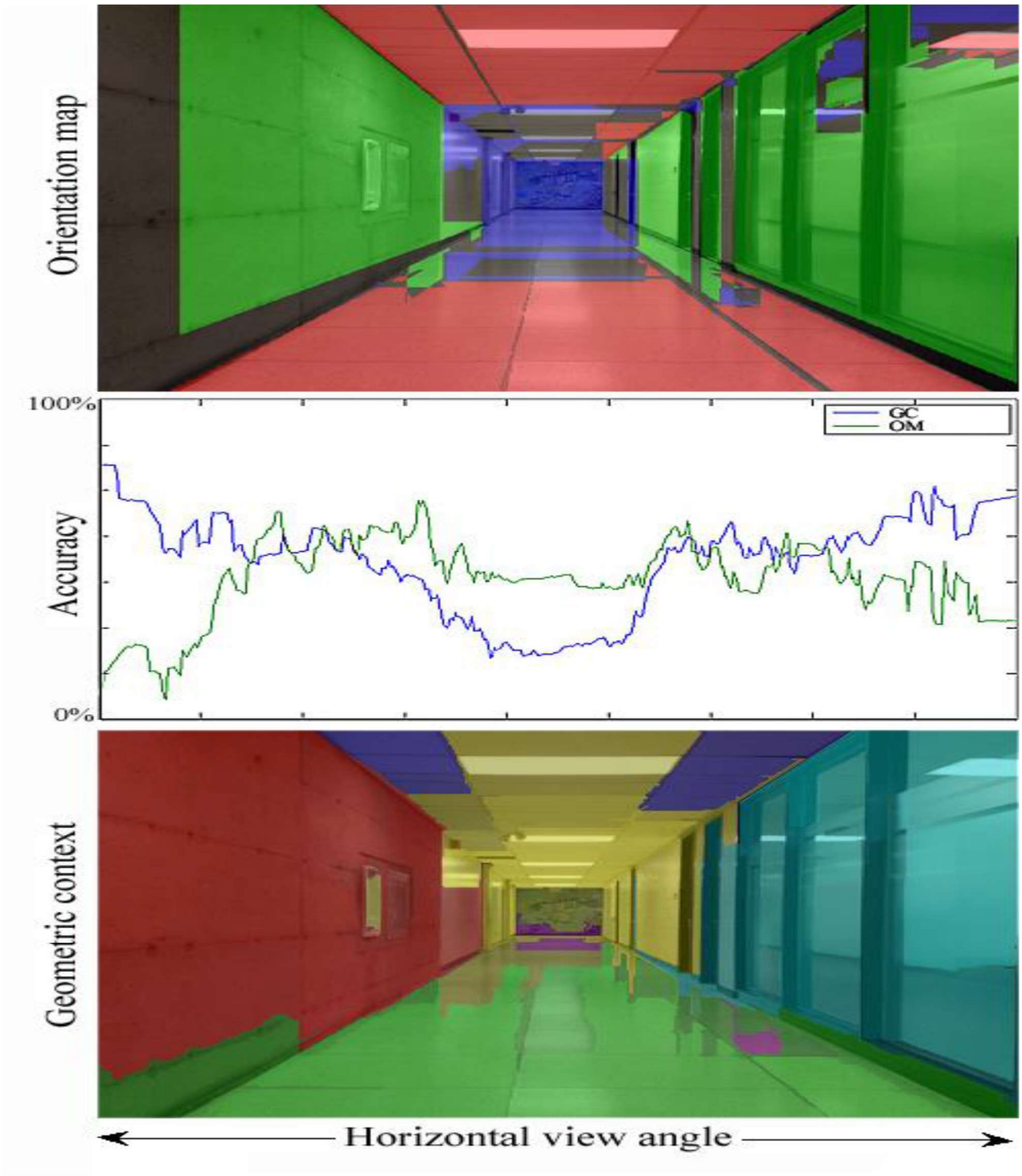


Figure 3.1 Orientation Map and Geometric Context accuracy changes by changing the horizontal viewing .From [67]

From figure 3.1 when moving from left to right the accuracy of GC and OM calculation change fundamentally. GC calculation performs better than OM on both right and left edge while OM perform better on the middle the picture [64].

Geometric context method are first proposed by Hoeim [69] for outdoor environment .Hoiem [69] et al consider surface estimation as a recognition problem and they are able to recover the coarse surface layout in a wide variety of outdoor scenes. Their approach has the advantages of simplicity and robustness, being able to generalize even to paintings and indoor images. One important aspect of their approach is the use of a wide variety of image cues including position, color, texture, and perspective. Different cues provide different types of information about a region, and, when used together, they are quite powerful. The idea of multiple segmentations is also crucial to the success of their algorithm, especially for distinguishing among the subclasses. But their method has limitation for indoor scene Hedau [69] et al have improved Hoeim [69] algorithms significantly. On a dataset of 204 training and 104 test images of indoor scenes, Hedau [69] et al report that the pixel error in surface labels drops from 26.9% to 18.3% when considering the box layout.

Additionally, the features from the surface label estimates improve the estimates of the box labels. When measured as the pixel error in the wall/floor/ceiling regions, error drops from 26.5% to 21.2%. When measured as RMS distance between predicted and true corners, error drops from 7.4% to 6.3%, as a ratio of the image diagonal length.

3.5 Development Tools

The research is developed using the following development tools:

- Ubuntu 16.04.2.1 LTS operating system
- oracle virtual machine(VM) virtual box 6.1.
- Mat Lab 2017

3.5.1Image labeling tool

To prepare images for training appropriately labeling and organizing the training images is important. For these research the following image labeling tools are utilized

3.5.1.1GIMP (GNU Image Manipulation Program)

GIMP is a free and open-source designs editorial manager utilized for picture modifying and altering, freestyle drawing, changing over between various picture format, and progressively particular assignments.

GIMP started life during the 1990s as the GNU Image Manipulation Program, and the free, open-source picture altering instrument has developed in both in multifaceted nature and usability after some time. The most recent variant, GIMP 2.8, keeps up the program's heritage as an incredible and exceptional, yet absolutely free picture editor, it's a paint and drawing device, a photograph re toucher, and a group handling and change instrument, across the board, with complex highlights layers, filters, and effects.

Gimp design is consistent, instinctive, and even appealing yet it additionally accompanies huge amounts of help; from various Help records (Help; Context Help; User Manual) and significant online assets, for example FAQs, documentation, tips, source code etc.

3.5.1.2 Ground Truth Labeler

The Ground Truth Labeler application gives a simple method to check rectangular locale of interest (ROI) marks, polyline ROI names, pixel ROI names, and scene names in a video or picture arrangement.

Ground truth labeler permits the client to:

- Manually mark a picture outline from a video.
- Automatically mark across picture outlines utilizing automation algorithm.
- Export the marked ground truth information.

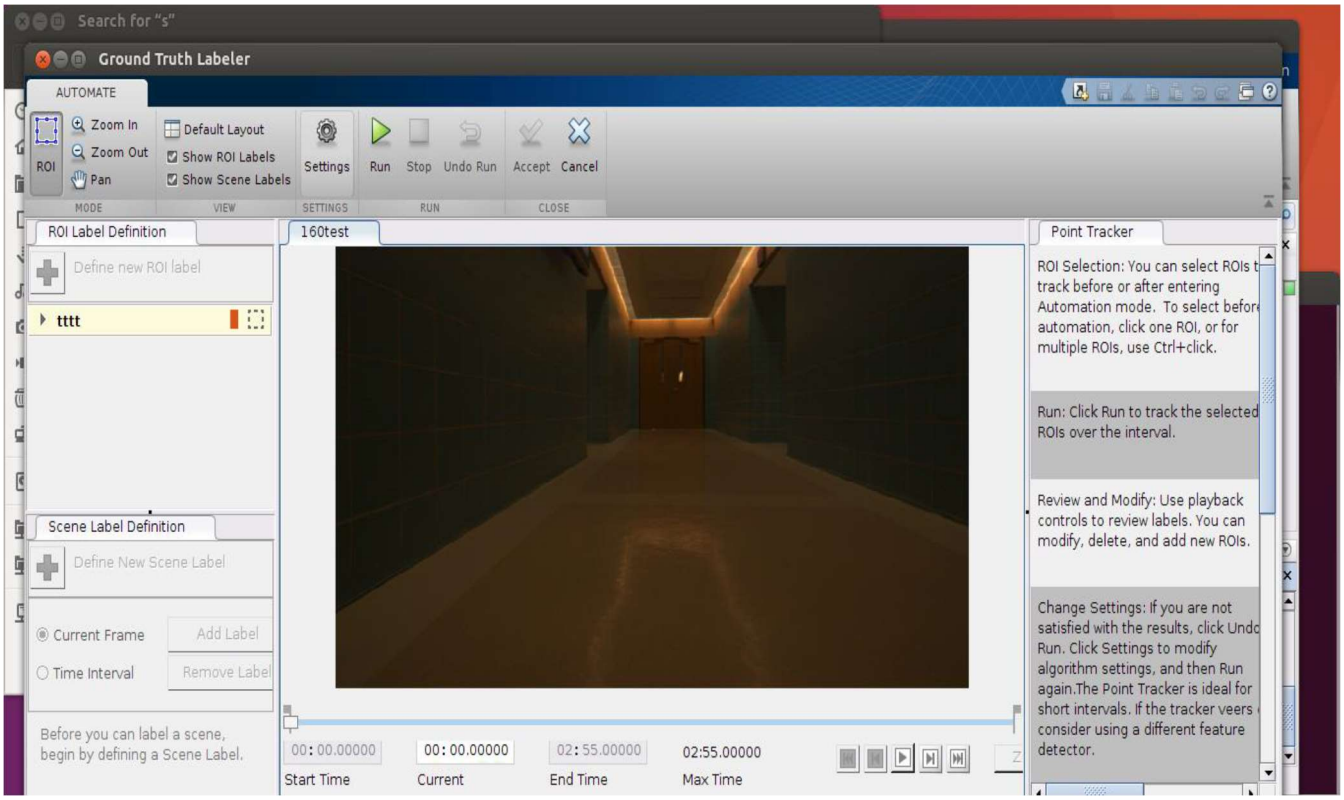


Figure 3. 2 ground truth labeler

3.5.2 3D visualization tools

To create the model of the indoor environment and visualize the 3d structure I use visualization tools. Moreover, we can view the effect of creating map and navigation using the tools mentioned below

3.5.2.1Gazebo

Gazebo is a robot simulation system. Gazebo empowers a client to make complex environments and offers the chance to simulate the robot in the model made by the user. In Gazebo the client can make the model of the robot and join sensors in a three-dimensional space. For the model of the robot, the client can utilize the URDF document and can use joints to the robot.

3.5.2.2 Rviz

Rviz is a simulation system where we can see the sensor information in the 3D environment. For instance, using the laser scan information, it is possible to manufacture a map and utilize that for auto navigation. In Rviz it is possible to get and graphically represent the values utilizing camera picture, laser filter and give the level of movement for each piece of the robot and so forth.

3.6 Development Platforms

To implement the proposed research, I will use the following open source robot platforms

3.6.1 Robotic Operating System (ROS)

ROS is an open-source framework planned to be utilized with a robot. It offers the types of assistance that are normal from operating system, similar to hardware abstraction, low-level device control, execution of generally utilized functions and message-passing between processes. Ros gives an assortment of tools, libraries, and conventions with the reason that creating powerful robotic behavior in various robotic platforms[72].

Robotic Operating System (ROS) is a free and open-source and one of the most popular middle wares for robotics programming. ROS comes with message passing interface, tools, package management, hardware abstraction etc. It provides different libraries, packages and several integration tools for the robot applications. ROS is a message passing interface that provides inter-process communication so it is commonly referred as middleware. There are numerous facilities that are provided by ROS which helps researchers to develop robot applications.

3. 6.1.1Simultaneous Localization and Mapping (SLAM)

SLAM is the computational problem of building or refreshing a map of a new unexplored region while simultaneously monitoring the robot area inside it. ROS have a gmapping bundle, which gives laser-based SLAM (Simultaneous Localization and Mapping), as a ROS node called slam gmapping. Utilizing slam gmapping, a 2D map can be made (like a structure floorplan) from a laser and posture information gathered by a versatile robot

3.6.2 TurtleBot3

TurtleBot3 is a little, inexpensive, programmable, ROS-based versatile robot for use in training, exploration and prototyping. The TurtleBot3's main innovation is SLAM, Navigation and Manipulation, making it reasonable for home assistance robots. The Turtle Bot can run SLAM algorithms to make a map.

Turtlebot3 is robot which uses raspberry pi3 and Arduino board for operating and controlling the robot. This robot has LIDAR which is mounted at the top which is used to measure distance from reflected object as well as to useful to collect landmarks. And also we can integrate camera to raspberry pi3 with initial port for transferring the image and be able to processed it on the computer. Turtlebot3 also has DYNAMIXE two rotating wheel that can measure the odometery which is useful data to measure the position of the robot by its rotation.

I select turtlebot3 because of the following features

- Compatible with ROS Platform

Turtle Bot is the most popular open source robot for education and research. The new generation TurtleBot3 is a small, low cost, fully programmable, ROS based mobile robot. It is intended to be used for education, research, hobby and product prototyping.

The Turtle Bot brand is managed by Open Robotics, which develops and maintains ROS. Turtle Bot can be integrated with existing ROS-based robot components.

- Open Source

The hardware, firmware and software of TurtleBot3 are open source which means that users are welcomed to download, modify and share source codes.

3.7 Evaluation Method

The result will be analyzed to describe the performance of scene analysis model on a test data set. The dataset is split into different training and testing set. The algorithm is evaluated using the test set The performance of the proposed algorithm is evaluated using average precision, recall, false positive per image and log average miss rate.

CHAPTER FOUR

4. Proposed Work

4.1 Overview

The main aim of this research is integrating scene analysis especially room layout recovery algorithm with object detection to improve indoor robot navigation. Even though there are plethora of research that deal with scene analysis and object detection separately, to my knowledge research's that tries integrating these algorithms to improve robot navigation are few.

Trying to integrate room surface layout recovery with object detection make more sense because of the following reasons. Consider if a robot has a prior knowledge of the surfaces, for example, floor, wall, celling then it is obvious that a door is located on the wall not on the celling or floor. The same applies for text and sign. This would remarkably reduce the search space for object detection algorithm. If the algorithm is trying to detect a door or text, then instead of searching the whole image it would limit the search space only to the walls because you can't find doors on the ceiling. More over this improves robot navigation by uniquely identify rooms, there by solving Loop closure problem. Loop closure problem is robot is in ability to recognize a place it has already visited because sensor and measurement error. Additionally, Obstacles are easily identified because room layout algorithm differentiate between the floor and the obstacle or object placed on the floor.

Taking insight from cascading classifiers, a form of ensemble learning, in which the output of one classifier is used to train and improve another classifier. I propose to integrate room layout recovery algorithm with Aggregate Channel Feature(ACF) object detection algorithm. First I take the original image and apply room layout recovery algorithm, to recover and segment the walls (right, left and center). floors, and ceiling. Then I label the segmented image for door, text and sign. After that the labeled image is used to train ACF object detector to detect doors, text and sign which improve the accuracy of ACF detector and decrease the search space and also uniquely identify rooms by text and sign on the doors.

The proposed integration of room layout recovery algorithm with Aggregate Channel Feature(ACF) object detection algorithm is explained below in detail.

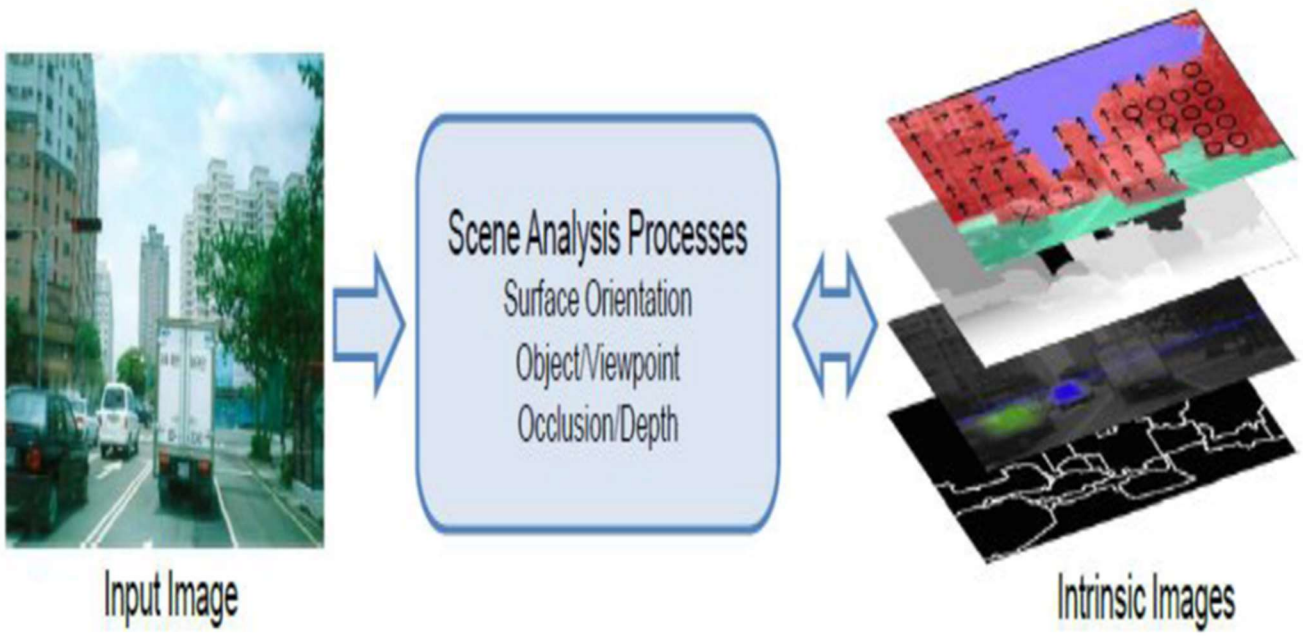


Figure4. 1 scene analysis from[73]

4.2 Scene analysis

For scene analysis a geometric context(GC) based room layout recovery algorithm originally proposed by Hoeim et al [69], [29] with some modification and improvement by Hedau et al [5] is selected. Hoeim et al [69], [29] build on idea of geometric sketches to a general representation of the scene in terms of what is called intrinsic images. Intrinsic image is a map that represents a one property of a scene. For example, the maps could indicate surface orientations, object boundaries, depth, or shading.

This algorithm is employed for these study because

- Primarily, the algorithm is modular, permitting another procedure to be embedded without upgrading the whole system. Therefore, it can easily accommodate new types of information. Like additional detectors for sign and text.
- Additionally, by permitting one procedure to impact another through signals, as opposed to hard constraints the algorithm is robust and not exposed to researcher-designed models.so it can extend to new environment.

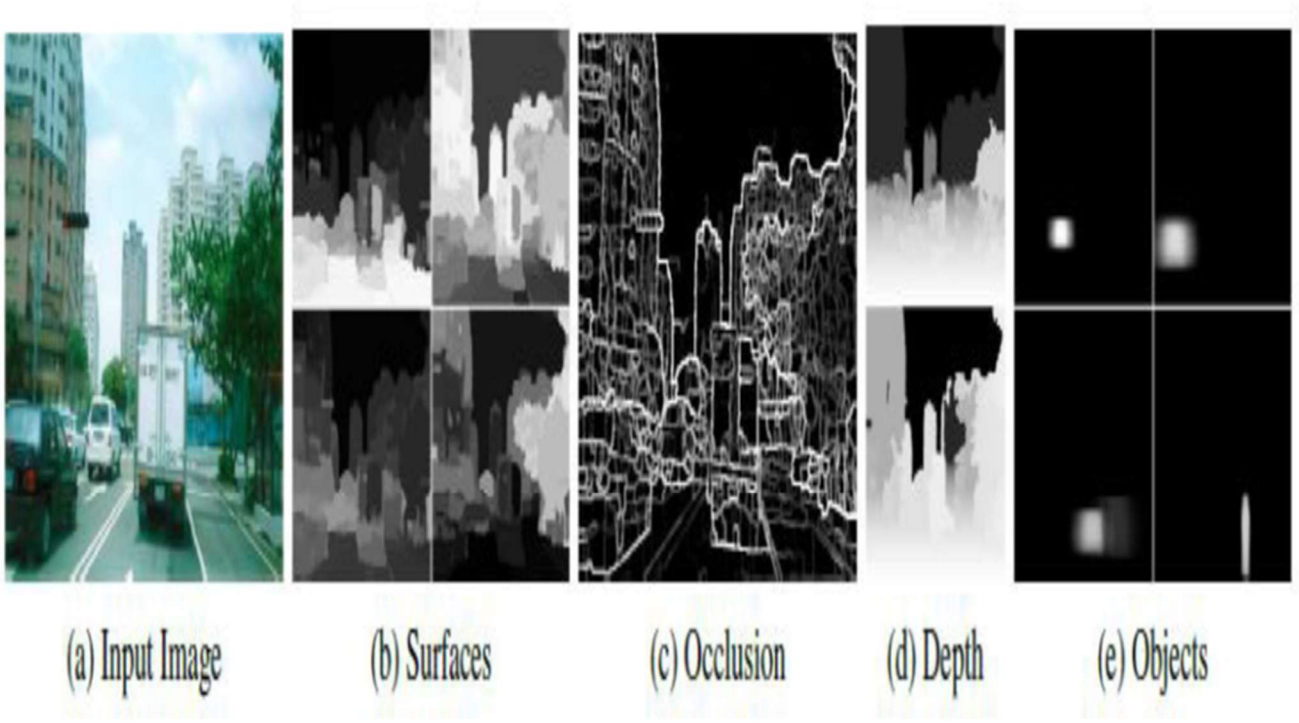


Figure 4.2Instances of intrinsic images assessed from the principal image (a) in (b), the four major surface confidence maps are shown brighter means greater confidence; clockwise, from upper-left: —support‖, —vertical planar‖, —vertical porous‖, —vertical solid‖. In (c), certainty map for occlusion borders brighter means greater confidence in occlusion. (d), shows upper and lower estimations of depth brighter means closer). (e), depicts four of the major object in images. Every one of them are certainty map demonstrating the probability of every pixel having a place with an individual object (vehicles or people on foot for this situation) From [69]

4.2.1Intrinsic Images

Every intrinsic image is a map that represents a scene. These intrinsic images mirror the probabilities of the approximations, either by representing the probabilities directly, likewise with the surfaces, or by including a few assessments, as with the depth Surfaces [73].

The surface images consist of seven confidence maps for —support‖ (e.g., the ground), vertical planar facing —left‖, —center‖, or —right‖ (e.g., building walls), vertical non-planar —porous‖ (e.g., tree leaves), vertical non-planar —solid‖ (e.g., people), and —sky‖.

Each image is a confidence map for one surface type indicating the likelihood that each pixel is of that type. The intrinsic images are computed using the algorithm described below generally there are three main steps

First the image is partitioned several times into multiple segmentations.

Then Image cues are then computed over each segment,

and finally a boosted decision tree classifier estimates the likelihood

4.2.2 Approach to Estimate Surface Layout

The contrast among surfaces and objects is that an objects will in general relate to a specific sort of surface for instance, as observed from in figure 4.2 the street is a supporting surface, and a person on foot is a vertical, non-planar solid surface. Moreover, numerous objects, such as vehicles and people tend to rest on the ground, so a visible supporting surface lends evidence to a hypothesized object.

The values of the surface maps near a detection region can strengthen or weaken the confidence in the detection (e.g., if the area below the detection region is a supporting surface, confidence should increase). Likewise, because detected objects have a known geometry for instance vertical non-planar solid. The pixel labels of the objects can provide valuable cues for geometry estimation. Likewise, viewpoint estimates recovered from detected objects can improve surface estimates [69].

Objects have a structured arrangement in well-formed scenes. Beds cannot stick through walls. People usually do not hang chairs from the ceiling. Chairs are placed on the floor, often near tables. objects are arranged according to physical laws and for convenience in physical interaction. We can best take advantage of their organization with some knowledge of the physical scene space.

Hedau [5]et al. provide examples of how to incorporate some simple layout principles, such as that objects cannot stick through walls and that beds are likely to have one side near a wall. These detections, in turn, are used to refine estimates of the ground plane

Every region in the image is categorized into one of three main geometric classes or surfaces: ―support‖, ―vertical‖, and ―sky‖. Support surfaces are roughly parallel to the ground and could potentially support an object. Examples include road surfaces, lawns, dirt paths, lakes, and table tops. Vertical surfaces, which are defined as too steep to support an object, include walls, cliffs, curb sides, people, trees, or cows. The sky is the image region corresponding to the open air and clouds [69].

To provide further geometric detail, [69]divide the vertical classes in to several vertical sub classes: planar surfaces facing to the ―left‖, ―center‖, or ―right‖ of the viewer, and non-planar surfaces that are either ―porous‖ or ―solid‖. Planar surfaces include building walls, cliff faces, and other vertical surfaces that are roughly planar. Porous surfaces are those which do not have a solid continuous surface. Tree leaves, shrubs, telephone wires, and chain link fences are all examples of porous surfaces. Solid surfaces are non-planar vertical surfaces that do have a solid continuous surface, including automobiles, people, beach balls, and tree trunks.

The algorithm to predict geometric labels(support, vertical, sky) from an image is outlined below[73].

1. Split the picture into many little areas. The little locales, called "super pixels", offer spatial support for color and texture histograms. The graph-based algorithm is of utilized to create the super pixels.

2. Gather the super pixels into numerous bigger regions that offer better spatial support for perspective cues. Initially, a logistic regression algorithm is trained to forecast the probability that two super pixels have a place with a same surface, based on a comparison of their color, texture, and position. Then, groups are greedily made to make the most of this pairwise probability inside the groups. By shifting the number of cluster centers and the initialization, numerous segmentations are created.

3. Calculate features for every region. The feature set is intended to encode clues about the material like means and histograms of color and texture and surface positioning for example, texture, histograms of orientations and convergences of straight line sections, and histograms of edges allocated to vanishing points. Moreover, computing other region properties including size, shape, and position. For color features, the HSV and RGB color spaces are utilized, and texture responses are predictable with the LM filter bank.

4. Allocate the confidence that every region has a place with each geometric class utilizing boosted decision tree classifiers and the computed features. Likewise, allocate the certainty that every region relates to a single surface, utilizing similar features and strategy. The

classifiers are trained utilizing training images that have been produced using the similar multiple segmentation methodology. The boosted decision trees are exceptionally adaptable classifiers that yield scores dependent on combinations of chosen features.

5. Calculate the confidence of every geometric class for every pixel by averaging over the regions that encompass the pixel, weighted by the confidence of that the region corresponds to a single surface. Sometimes, it is preferable to keep the confidence value; other times, it is better to choose the label with the most confidence

SURFACE LAYOUT ESTIMATION
1. Image! Super pixels via over-segmentation
2. Super pixels! multiple segmentations (a) For each super pixel: compute cues (b) For each pair of adjacent super pixels: compute same-label likelihood P(yi = yj |I) (c) Create multiple segmentations for varying ns
3. Multiple segmentations! Super pixel labels (a) For each segment: i. Compute cues (Table 1) ii. For each possible label (main classes and subclasses): compute label likelihood P (˜yj |I, sj) iii. Compute homogeneity likelihood P(sj |I) (b) Compute label confidences for each super pixel: P(yi|I) /Xsj3i P(˜yj |I, sj) P(sj |I)

Table 4.1surface estimation

4.2.3. Pseudo code to train Surface Layout algorithm

Outline of training procedure [9, 69] to recover surface label from single image

Input: Image

1. For each training image:

    (a) Compute super pixels

    (b) Compute super pixel cues

2. Train same-label classifier

3. For each training image:

    (a) Produce multiple segmentations for varying number of segments

    (b) Label each segment according to ground truth

    (c) Compute cues in each segment

4. Train segment label classifier and homogeneity classifier

4.2.4. Recovering room layout

The algorithm described above would recover surface layout of outdoor environment but for indoor environment it is also important to take advantage of the structure of indoor environment, not only recovery the surfaces (walls, floor and ceiling for indoor environment) but also grasp the entire space in the room. To that end Hedu [5] modified Hoiem [66] algorithm for indoor structures. The algorithm models the room as box.

4.2.4.1 The room as a box

The image below shows the steps taken to recover free space and model the room as if it is abox

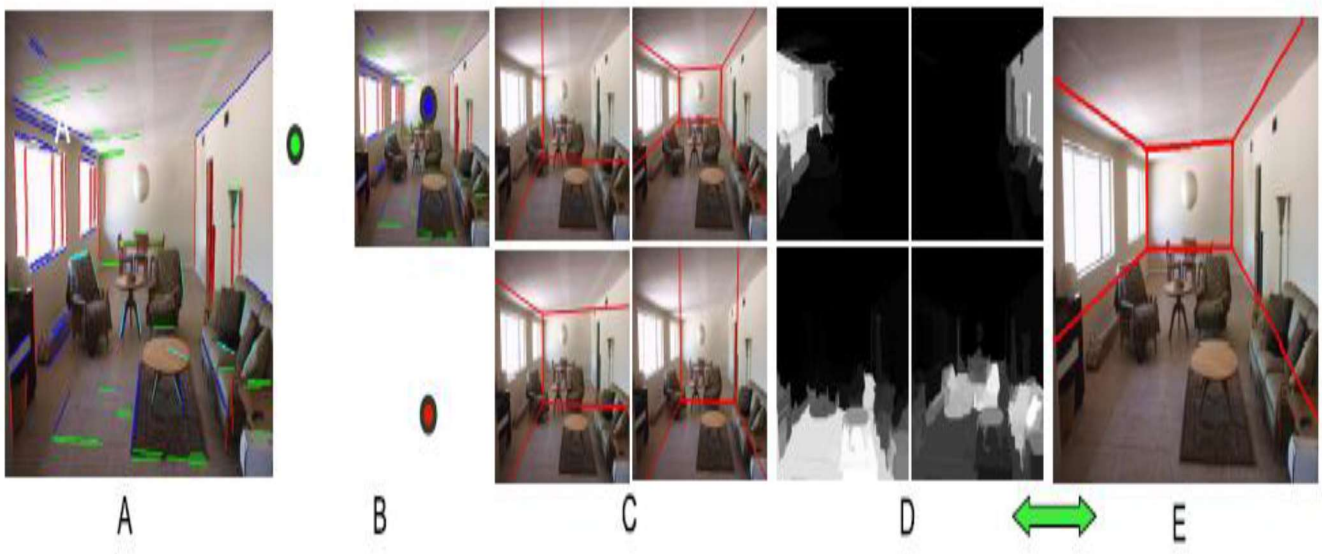


Figure 4.3 modeling the room as box from [5]

Since the aim is to model the visible surfaces of a room and the full extent of the walls. The earlier described method of Hoiem et al [66] will not work well for indoor scenes, because rooms are full of objects that obscure wall-floor boundaries.to overcome this shortcoming i employ the approach of Hedau et al [5] which model the scene jointly in terms of a 3D box layout and the surface labels of pixels. The box layout coarsely models the space of the room as if it were empty.

The surface labels, similarly to geometric classes from Hoiem et al [66], provide pixel localization of the visible object, wall, floor, and ceiling surfaces. To achieve robustness to clutter (objects in the room, obstacle in my case), three approaches are employed[5].

To start with, the powerfully parameterized 3D box model can be projected strongly from sparse visual evidence,

Then, the parameters are projected cooperatively using structured prediction founded on global perspective cues.

Finally, clutter is unambiguously modeled through the surface labels, so that misperception due to clutter can be avoided when estimating the box.

Similarly, the 3D box approximation offers valuable cues to improve the surface label estimations.

4.2.4.1.1 Algorithm to recover room layout in 3Dspace

The algorithm to recover room layout in 3Dspace[5] is described below

I. Estimate three mutually orthogonal vanishing points (Figure 4. 3 A, B). The vanishing points specify the orientation of the box, providing constraints on its layout.

II. . Generate many candidates for the box layout (Figure 4. 3 C) by sampling pairs of rays from two of the vanishing points (Figure 4. 3 B).

III. Compute perspective cues for each box. The features are the fraction of edge pixels within each box face that have been assigned (in step I) to the face ‘s vanishing points

IV. Compute the confidence of each 3D box hypothesis using a linear classifier. The classifier is learned using structured learning to minimize the error in predicting the corners of the box and to minimize the overlap of the predicted wall/floor/ceiling regions with the true ones in the training set

V. Estimate the surface labels given the most likely box candidate. The surface layout algorithm and features Hoiem [66] are used, with the addition of features that tell what fraction of each region overlaps with the walls, floor, and ceiling of the box layout. Rather than storing the most confident label, confidence maps are stored that indicate the likelihood of each surface label for a pixel (Figure4. 3D).

VI. Re-estimate the box layout using features from the surface labels (Figure4. 3 E). Features are added that indicate the average confidence of each surface label within each box face. Also, to improve robustness to clutter, new perspective features are computed that are weighted by the confidence of object labels (edges within likely object regions have small weight).

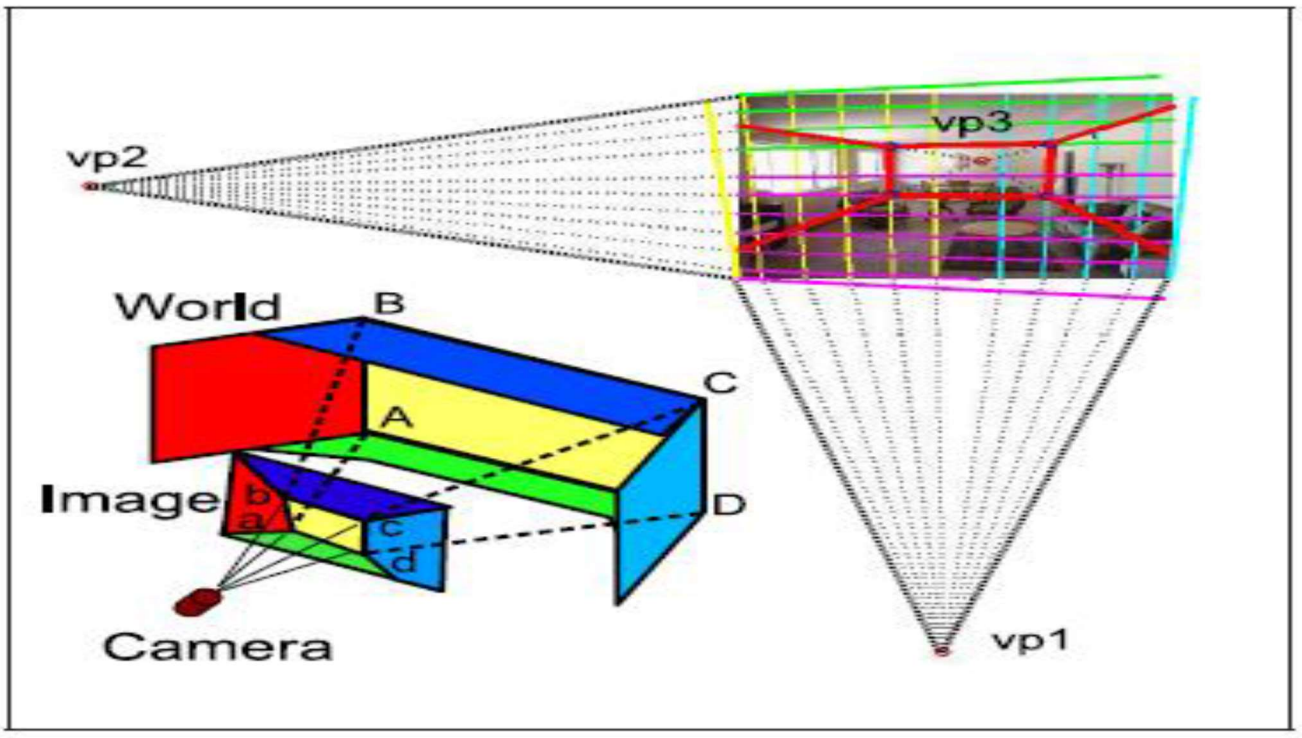


Figure 4.4 layout generation from [5]

4.2.4.1.2 Getting the Box Translation

Knowledge of the box orientation imposes strict geometric constraints on the projections of corners of the box, as shown in Fig. 4.4 and listed below. At most 5 faces of the box, corresponding to 3 walls, floor and ceiling, can be visible in the image, each projecting as a polygon. The 3D corners of the box are denoted by *A*, *B*, *C*, and *D*, and their counterparts in the image are *a*, *b*, *c* and *d*.

The vanishing points corresponding to three orthogonal directions in world are given by *vp*1, *vp*2, and *vp*3.[5].

1. Lines *ab* and *cd* should be co linear with one of the vanishing points, say *vp*1,

2. Lines *ad* and *bc* should be co linear with the second vanishing point, *vp*2, and,

3. The third vanishing point, *vp*3, should lie inside the quadrilateral *abcd*.

To generate the candidate box layouts, choose *vp*1 and *vp*2 as the two farthest vanishing points and draw pairs of rays from these points on either side of *vp*3. The intersections of these rays define

the corners of the middle wall, $a - d$ in the image. The rest of the face polygons are generated by connecting points $a - d$ to $vp3$. When fewer than 5 faces are visible, the corners will lie outside of the image, handled by a single dummy ray not passing through the image. An example box layout is overlaid in red in Figure. 4. 4

4.3 Aggregated Channel Features (ACF)

After recovering the room layout, the next step is training an object detector that can detect doors, text and sign. For this study I use Aggregated Channel Features (ACF) object detector. Aggregated Channel Features (ACF) is a form of object detectors called Integral Channel Features (ICF).

Integral Channel Features (ICF) is a method for object detection in that uses integral images to extract features such as local sums, histograms and Haar-like features from multiple registered image channels. This method was improved by Dollár *et al*. [74]in their work for pedestrian detection that introduce Aggregated Channel Features (ACF). Both ACF and ICF use the same channel features and boosted classifiers; the key difference between the two frameworks is that ACF uses pixel lookups in aggregated channels as features while ICF uses sums over rectangular channel regions (computed efficiently with integral images). Multiple registered image channels are computed using image linear/non-linear transformations, called integral channel features; ICF naturally integrate heterogeneous sources of information, have few parameters, and result in fast, accurate detectors.

4.3.1 Aggregated Channel Features (ACF) algorithm

The algorithm for ACF is described below Dollár *et al* [74].

1. Compute multiple registered image channels from an input image, using linear and non-linear transformations

2. Extract features such as sums over rectangular channel regions from each channel. The features extracted from various channels are called integral channel features.

3. Train the AdaBoost classifier. Dollár *et al.* used boosting technique which offers faster learning but training could be done with any of the other available methods such as support vector machine.

4. Finally, trained classifier is used to detect objects

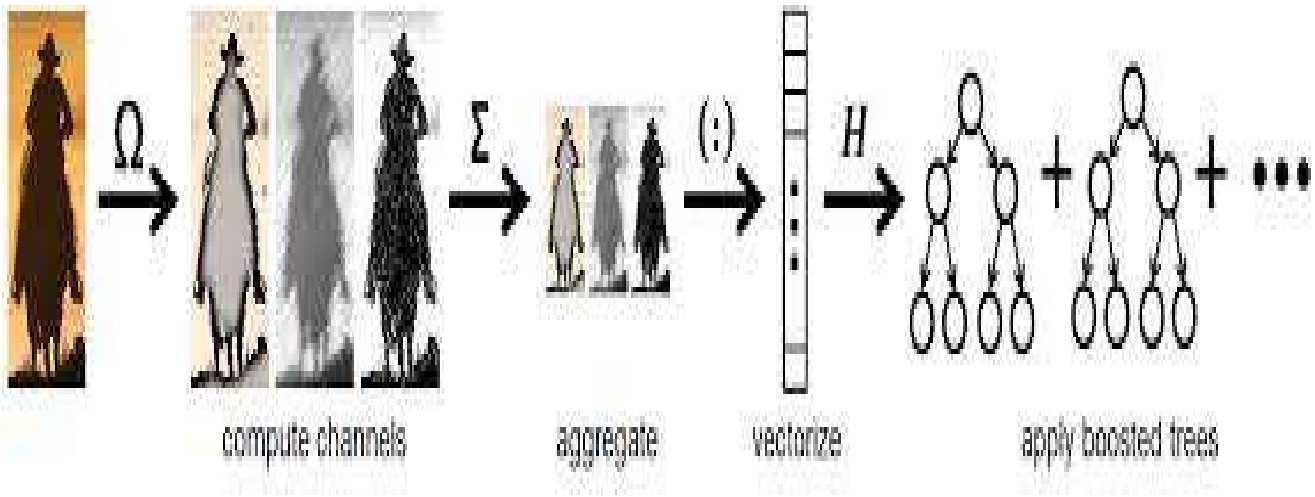


Figure 4.5 Overview of the ACF detector. Given an input image I, we compute several channels C = (I), sum every block of pixels in C, and smooth the resulting lower resolution channels. Features are single pixel lookups in the aggregated channels. Boosting is used to learn decision trees over these features (pixels) to distinguish object from background. With the appropriate choice of channels and careful attention to design, ACF achieves state-of-the-art performance in pedestrian detection From Dollár et al.[74].

4.3.2 Pseudocode for ACF

Input: Image

compute several channels C = (I),

sum every block of pixels in C

smooth the resulting lower resolution channels

apply Boosting to learn decision trees over these features (Features are single pixel lookups in the aggregated channels)

distinguish object from background

Output: detected object

4..4 Proposed doors, sign and text detection

For this study combining room layout recovery algorithm and Aggregated Channel Features (ACF) algorithm used. First stage is training the room layout recovery algorithm to recover and segment the surfaces then instead of using raw image to train Aggregated Channel Features (ACF) detector the output of the room layout algorithm is used to train ACF object detector.

For training 1885 images from Michigan indoor dataset ,247 text and 253 sign images are used. To illustrate the process a door containing a sign in hallway is shown below. The process is the same for detecting sign, text and doors.

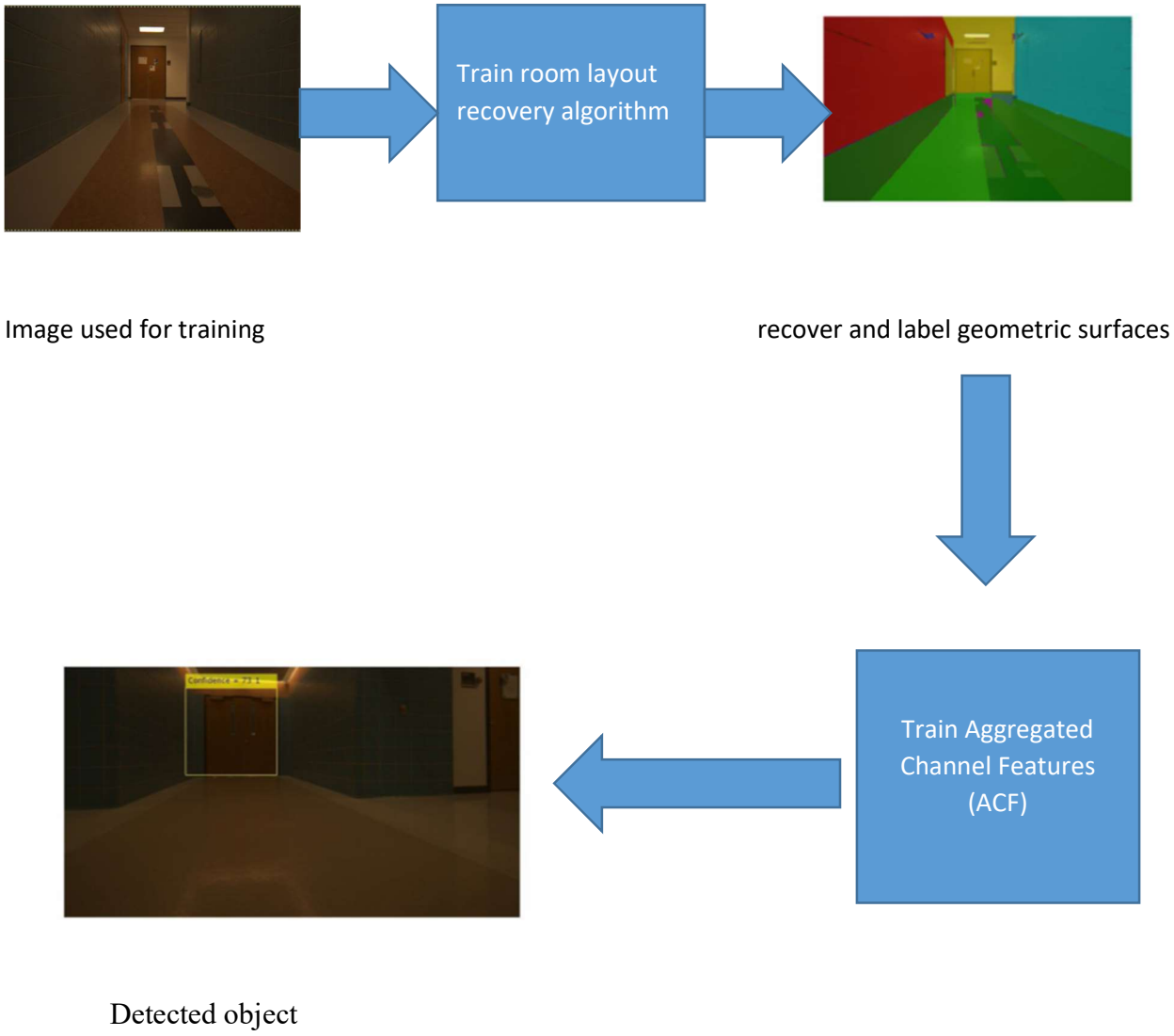


Figure 4.6 proposed object detection

4.5 Navigation

Navigation is important for a mobile robot to move from one point to another, avoiding situations like collision and unsafe conditions. The proposed Mobile Robot Navigation uses ROS Navigation

Stack with Turtlebot3, which supports SLAM. When building or navigating a map, SLAM ignores relevant descriptive information of the environment like what kind of object it contains. The proposed navigation is based on SLAM and object detection algorithm. The map creation, navigation process and how the object detection algorithm is integrated with navigation stack for navigation are described in the following subsections

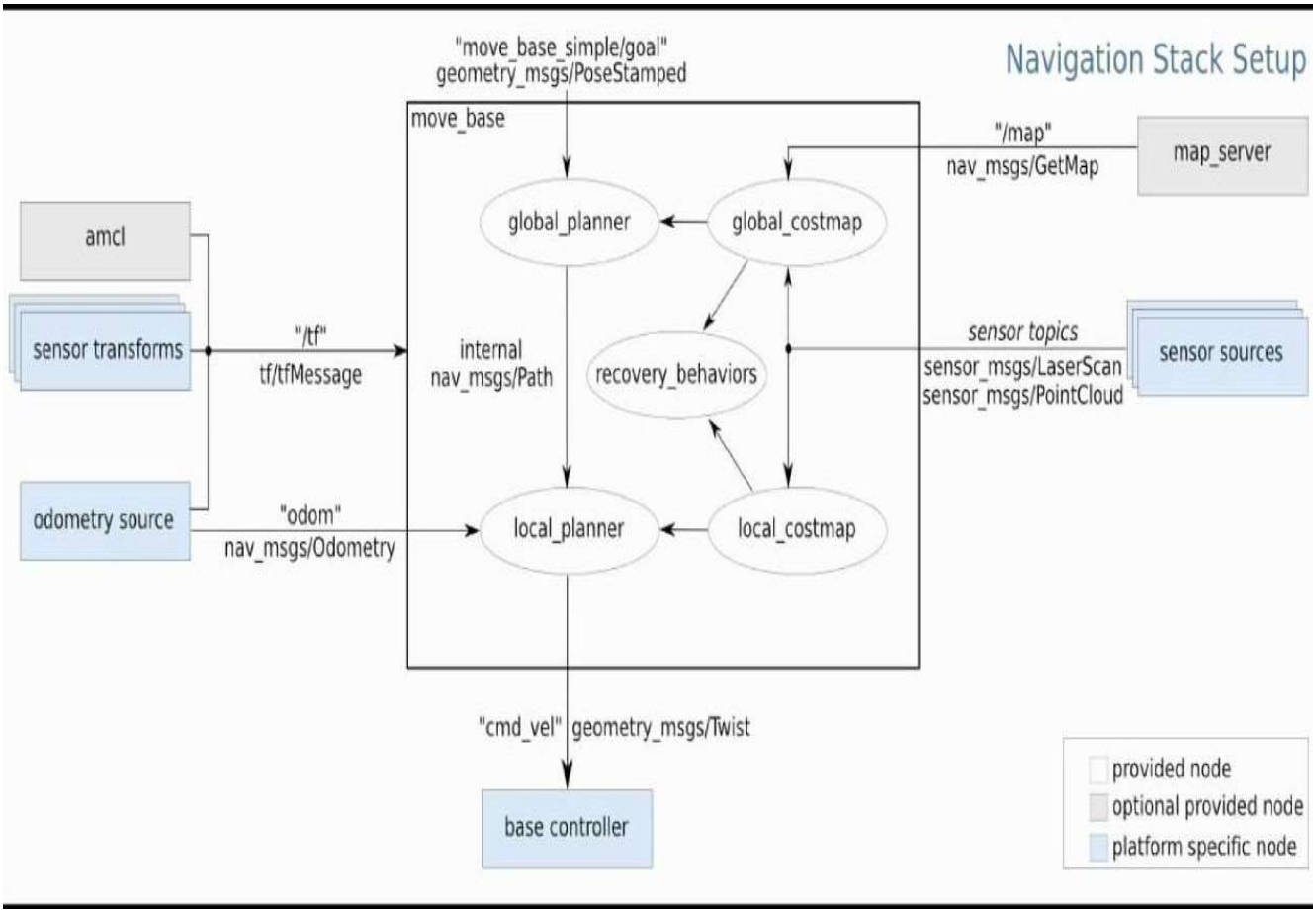


Figure 4. 7 navigation stack from [72]

4.6 SLAM Process

SLAM has the following vital steps described below

The first step is the robot takes landmarks as an input data using 360 degree routable LIDAR sensor. Landmark are features which can be re observed and differentiate easily from the surrounding and is used by the robot to estimate its own position

The second step is from the data input the features are extracted using spike landmark and RANSAC depending on the type of landscape. If the surface is not smooth it uses spike landmark to extract the feature point by finding values in the range of a laser scan where two values differ by more than a 0.5m. This means that to say there is landmark behind other landmark the laser has to return from the front landmark 0.5 m difference from the land mark existing behind. while RANSAC used for landscape which are smooth by finding lines from the extracting land marks and by randomly taking sample of laser reading it finds the best fit line that run through this reading by least square approximation method and checks for laser reading that lies close to this lines (e.g. edges).

The third step is during the journey of the robot it continually take up the features and it start to associate with the already stored data in the database. During the association if it found the new features it will store in the database and if it is already existing feature it will used to estimate the position of the robot by measuring the transition in the distance and angle to the old features.

Finally based on the new position and measurement data the map was updated and built tract the feature point by finding values in the range of a laser scan

Generally, to create a map using SLAM first by using LIDAR distance value have to be obtained from the X Y plane of the model object which is already built. Second the position of the robot has to be obtained by the principle of odometry sensor.

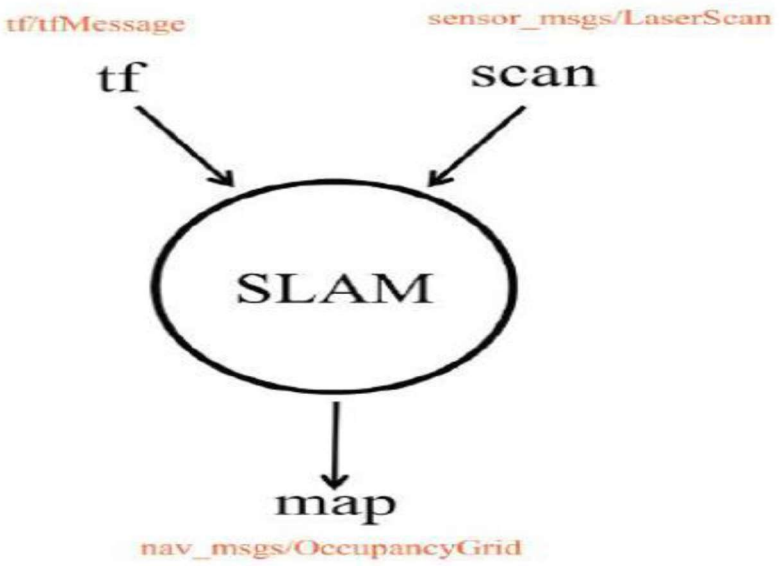


Figure 4. 8 slam message from[72]

In this flow each node is launched when launching the existing Launching SLAM file. Sensor node it runs the LIDAR sensor and sends the scan information for slam_gmapping node. In turtlebot3_teleop is teleop keyboard node that can receive the keyboard input and control the robot motion. This node sends the rotational and transitional speed command to the turtlebot3_core. Turtlebot3_core receives the transitional and rotational speed command and move the robot. In this node it frequently publishes the odom data which is going to be converted to relative coordinate in tf form. In turlebot_slam_gmapping node creates the map based on the scan information from the distance measuring sensor by LIDAR and the tf information. Finally the map saver node create map.pgn and map.yaml file which include the information file for the map.

Since the destination, the robot has to move is given by the user, the robot has to move to the destination automatically by observing sign and avoiding any obstacles. So navigation algorithm is responsible to receive data from the scene analysis and make decisions for the next move of the robot. It keeps track of goals and updates them when required. The image goals are sign and text. The navigation algorithm waits for receiving valid message. Once the message is received, it will tries to move the robot close to the goal and activate the path planning algorithm in ROS (cost map) to change the path previously it planned to move to.

CHAPTER FIVE

5. Implementation of room layout recovery and Aggregate Channel Features(ACF)

5.1 Overview

In this chapter, the implementation of the proposed room layout recovery and doors, sign and text detection is described. The implementation of room layout recovery algorithm and mobile robot navigation is described.

5.2 Prototype Development Setup

In these section the hardware, software and dataset used in these study are discussed.

5.2.1 Working Environment

Laptop:

- laptop computer is used for developing mobile robot navigation
- Oracle Virtual box 6.0
- Operating system: Ubuntu 16.04 LTS
- Processor: Intel ® Core™ i5-2540 CPU @ 3.60GHz x 4
- Installed memory (RAM): 4.00 GB (3.7 GB usable)
- System Type: 64-bit Operating System, x64-based Processor

ROS

- ROS kinetic kame 2016 is installed on the Desktop computer and Turtlebot3.

Mat lab

- Mat lab 2017 is used as a development IDE

Turtlebot3 Burger

- For simulation turtlebot3 is used for this study.

Gazebo and Rviz

- For mapping and visualization Gazebo and Rviz are used

5.2.2 Implementation Environment

An indoor room model is built for the purpose of mobile robot navigation in an indoor environment. The model represents an indoor environment. The environment is build using gazebo to simulate a typical room. Shown below

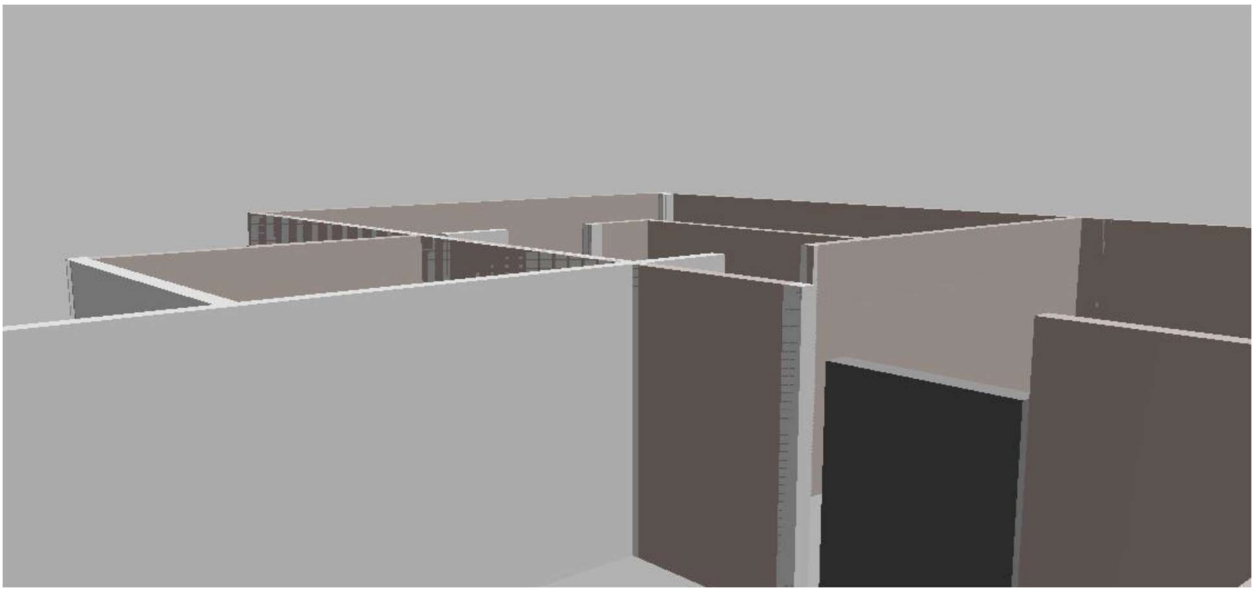


Figure 5.1 Implementation Environment 1

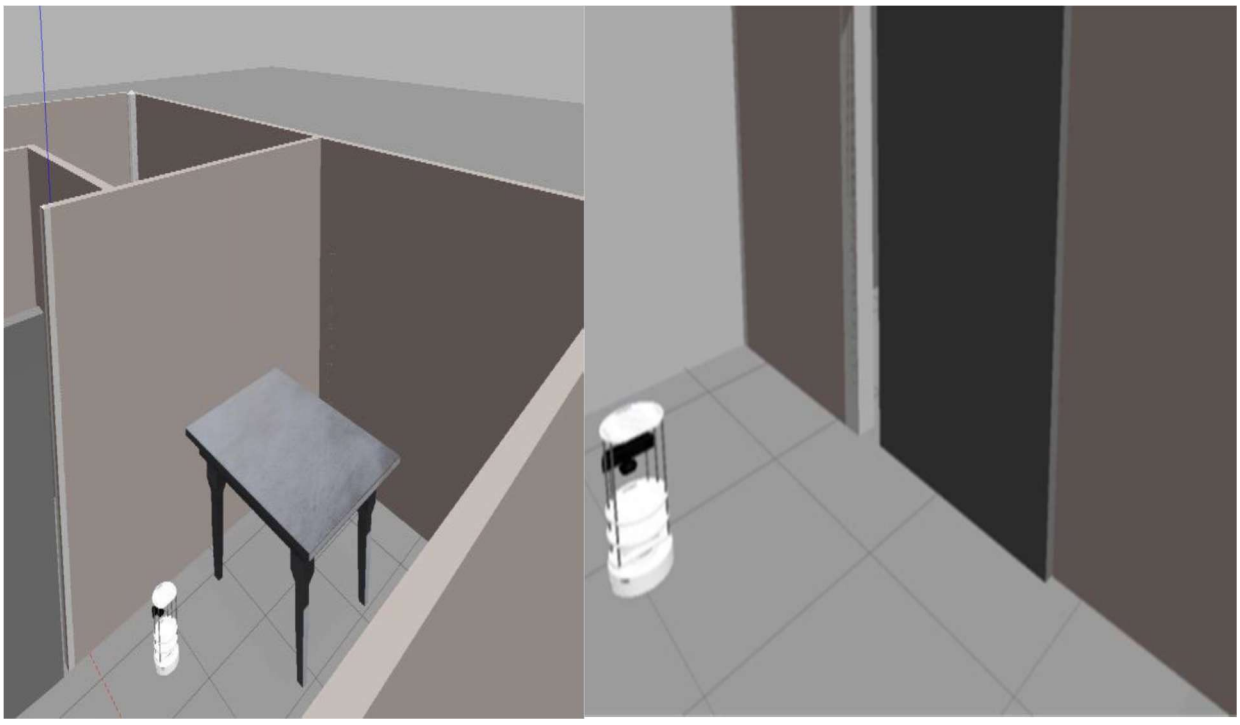


Figure 5.2 Implementation Environment 2

5.2.3 Dataset Description

The data set contains images of halls corridors, doors, text and signs. The dataset contains multiple images from different angles, scale, illumination, and view.

5.2.3.1 Michigan Indoor Corridor Dataset

The dataset [16] contains 4 video arrangements gained with camera mounted on a wheeled vehicle. The camera has a fixed tallness (0.47 m) with the ground all through the video. The camera they [16] utilized was: AVT Manta G-145 Color CCD Camera 5mm FL Wide Angle Low Distortion Lens. The camera was set-up so that there was zero tilt and roll concerning the ground. The intrinsic parameters of the cameras are: Focal length fc = [1389.182714 1394.598277] Principal point cc = [672.605430 387.235803]. The distortion of the camera has been amended The dataset gives a ground truth marking to all the pixels each 10 frames for every video. The labels can be deciphered as - 2 - > roof plane - 1 - > ground plane >0 - > walls [16].

5.2.3.2 Text and sign dataset

The text and sign dataset is collected from the internet and edited using Gimp

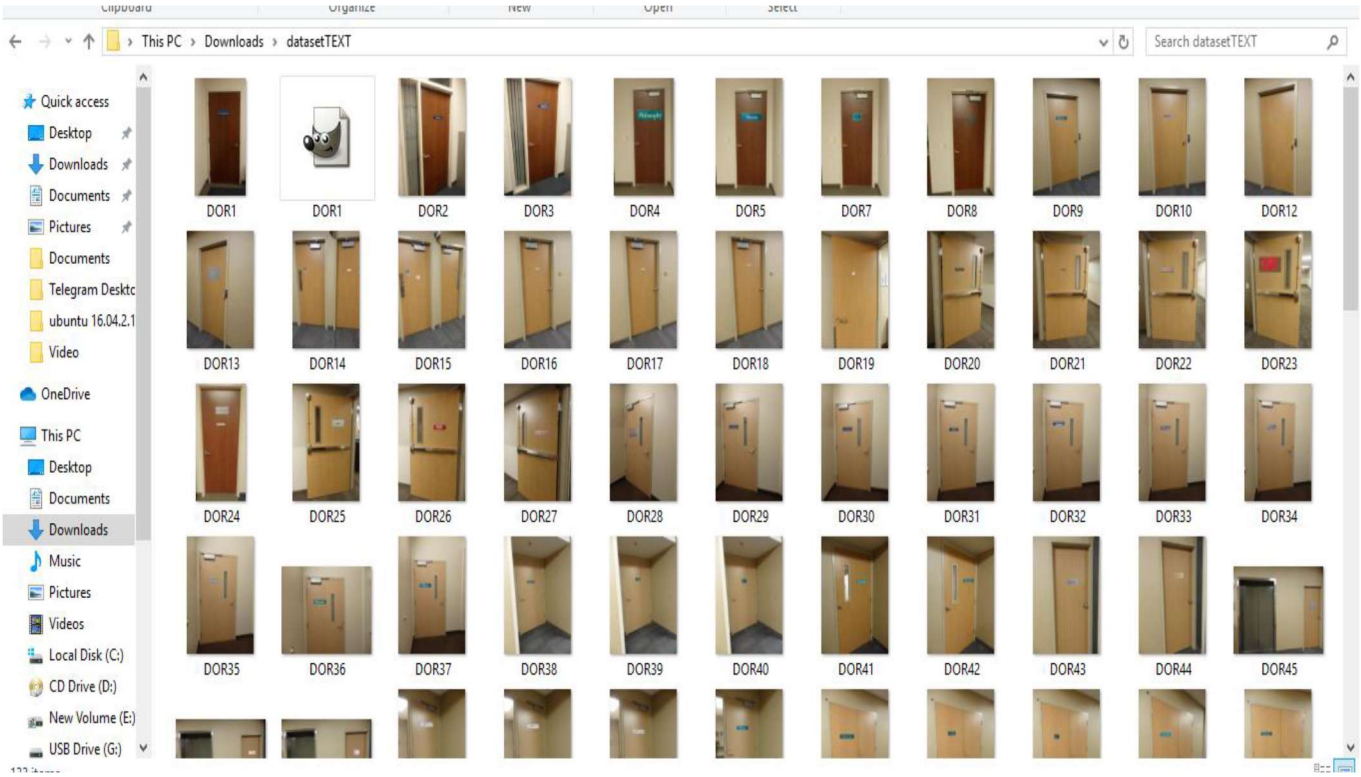


Figure 5.3 text dataset

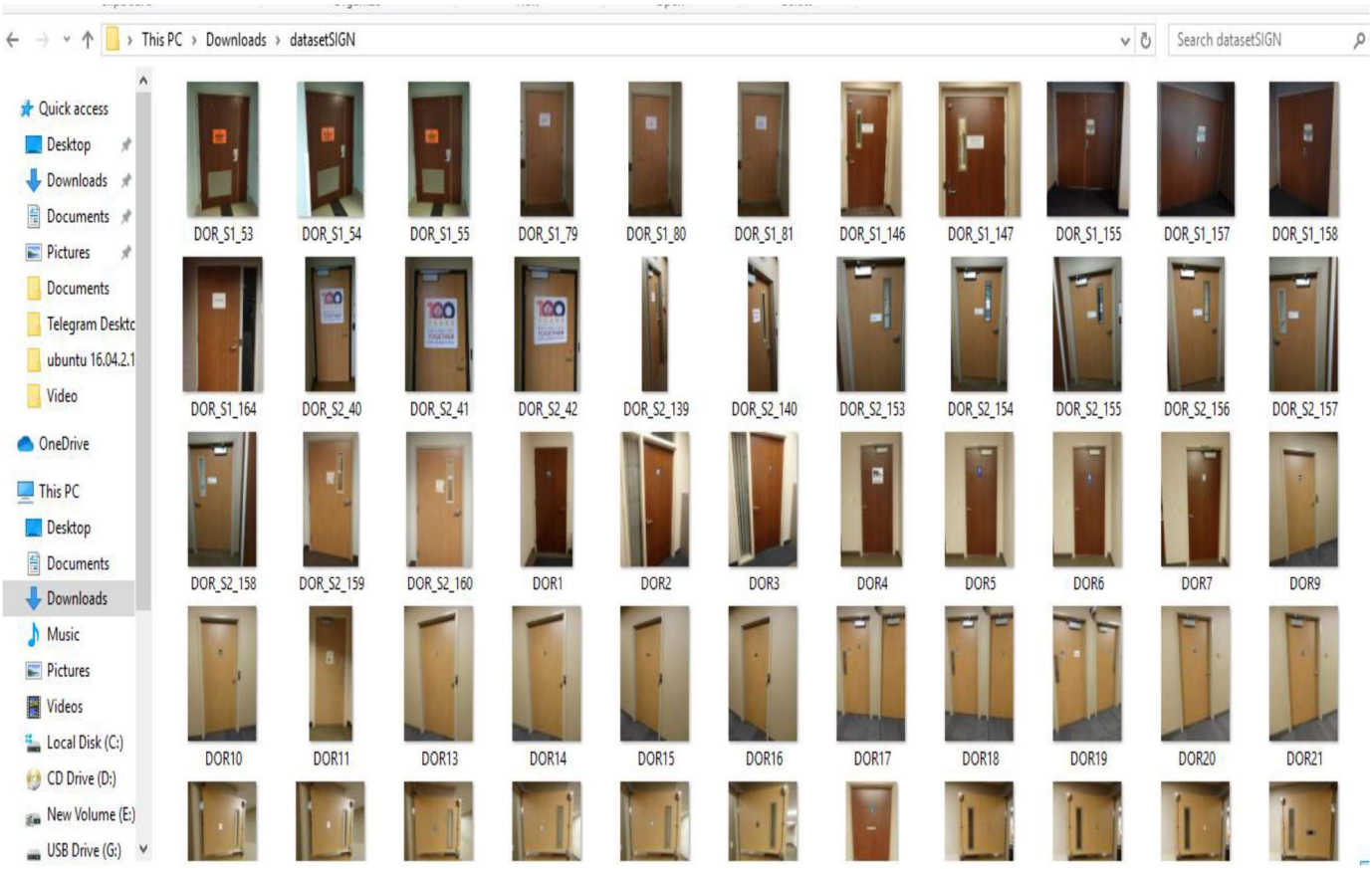


Figure 5.4 sign dataset

5.3 Implementation of major algorithms

The major steps are explained below. With sample code and images

5.3.1 Room layout recovery implementation

To implement the room recovery algorithm, I use matlab2017.The first step is labeling the training image from Michigan Indoor Corridor Dataset using ground truth labeler application of Mat lab.

To guide the reader through the implementation of the algorithm I show both the image and code below.

5.3.1.1 First step loads the training image

The first step is loading the training images .one of the images is shown below

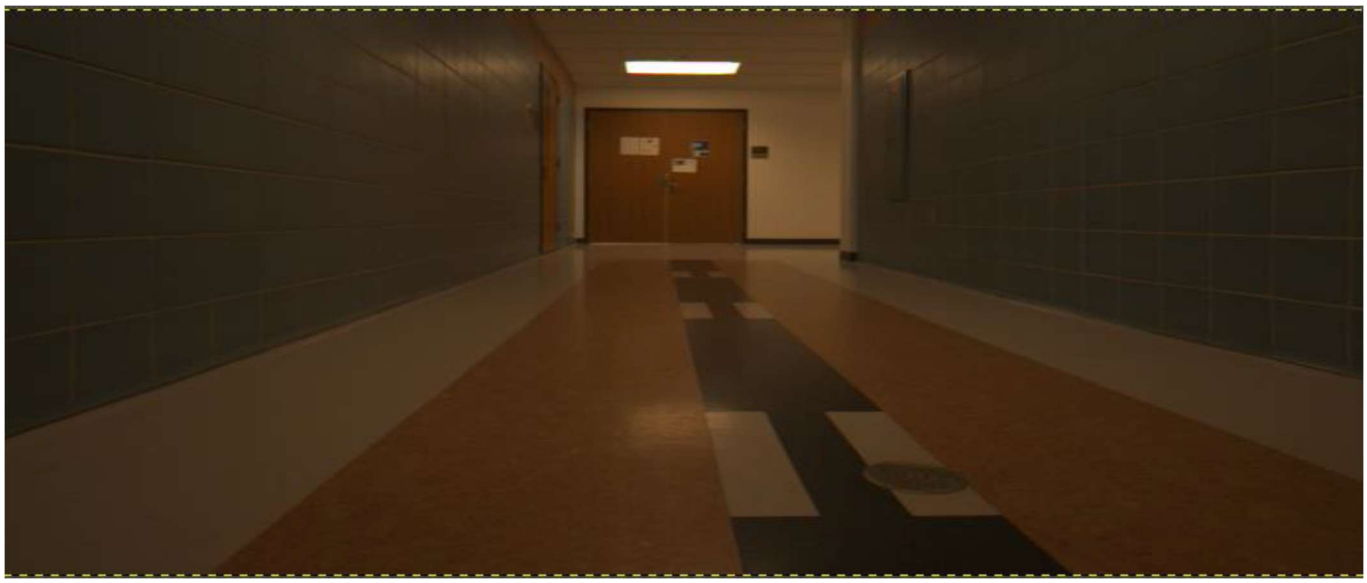


Figure 5.5 training image

5.3.1.2 Compute vanishing points

Indoor buildings have two principle qualities – a few lines in the scene are parallel, and various edges present are symmetrical. Vanishing points help understanding the indoor environment. Utilizing sets of parallel lines in the plane, the direction of the plane can be determined utilizing vanishing points [50]. As a result the first duty to recover room layout is calculating the vanishing points

```
[vp p All_lines] =getVP (imdir, imagename,0, workspcdir); // get vanishing points

img=imread([imdir imagename]);

[h w kk]=size(img);

VP=vp;

if numel(VP)<6            // number of vanishing point must be less than 6

return;

end
```

5.3.1.3 Get super pixel segmentation

Super pixel is a group of connected pixels with similar colors or gray levels. Super pixel segmentation is dividing an image into hundreds of non-overlapping regions. Instead of working with just pixels' super pixels can compute features on more meaningful regions. By computing super pixels, it is possible to find groups of pixels that are most similar to other groups and basically labeling those as being of the same type

```
nsegments= [5 15 25 35 40 60 80 100]; // number of segments

%fn= ['../Imsegs/' imagename(1:end-4) '.' segext];

fn=['/home/aklile/Documents/MATLAB/varsha_spatialLayout/SpatialLayout/Imsegs/'
imagename(1:end-4) '.' segext];

imseg = processSuperpixelImage(fn); // compute super pixels

tic

imdata = mcmcComputeImageData(im2double(img), imseg)

toc
```

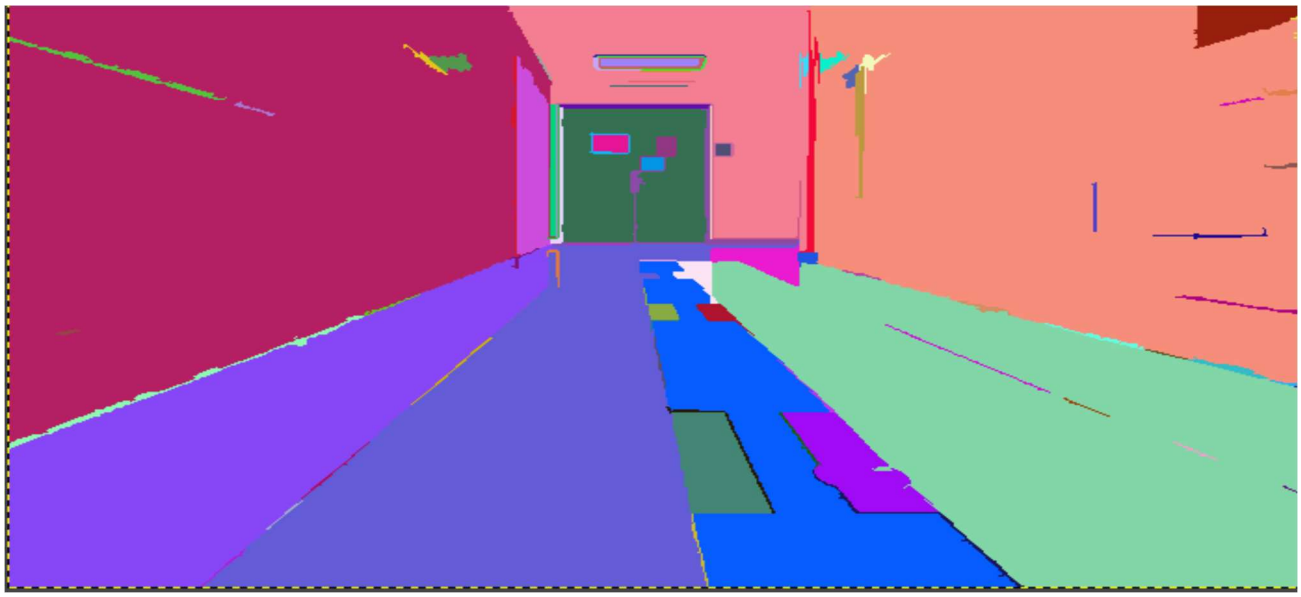


Figure 5.6 super pixel segmentation

5.3.1.4 Get surface label

There are 3 major surface labels which are walls (left, right and center), ceiling and floor. After determining which surface group a super pixel belongs the next step to segment the image with different color for each geometric surface.

```
normalize = 1;

pg=zeros (imseg. nseg,7); %

%get P(L/I)

pg = msTest imseg, segfeatures, smaps, ...

label classifier, segclassifier, normalize); //classify super pixels to geometric surfaces

filename=fullfile (workspcdir, [imagename(1:end-4) '_lc_gc.mat' ]);

save (filename, ‘pg')
```

5.3.1.5 Get candidate layouts

The next step is computing all the possible candidate room layouts that could be generated

```
[polyg, Features] = getcandboxlayout (vpdata. vp, vpdata.dim (1), vpdata.dim (2), integData);

for lay=1:25

layoutid=ii(lay);

Polyg=[];

for fie=1:5

Polyg{fie}=[];

if size(polyg{layoutid,fie})>0

Polyg{fie}=polyg{layoutid,fie};

end

tempimg=displayout(Polyg,w,h,img);

subplot(5,5,lay);imshow(uint8(tempimg),[]);title(num2str(vv(lay))); //draw candidate boxes
```

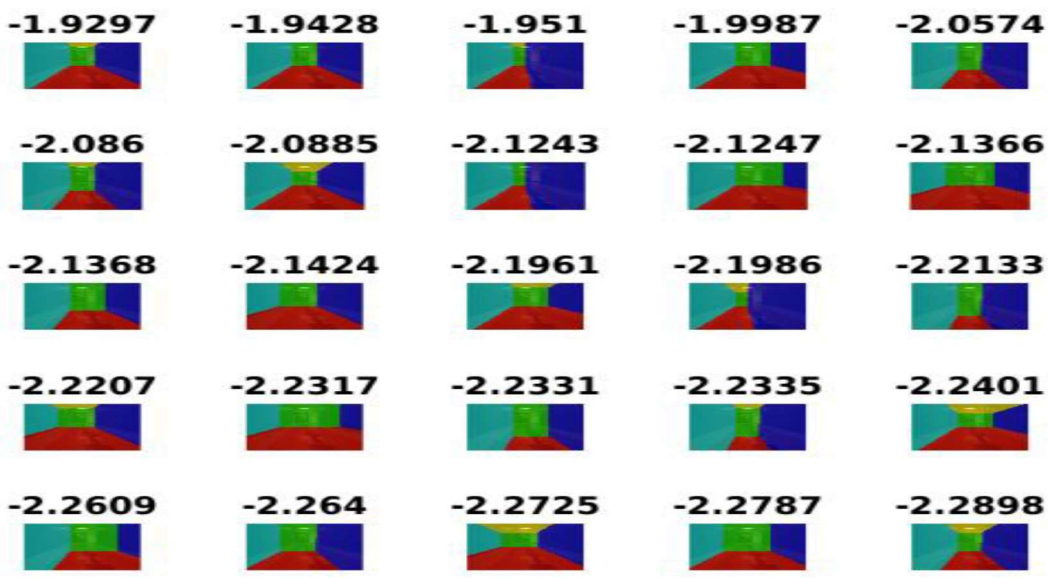


Figure 5.7 candidate box layout

5.3.1.5 Choose the best box layout

To recover free space in the room the next step is to draw box on the image that show the orientation of the room.

```
Polyg=[];

for fie=1:5

Polyg{fie}=[];

if size(polyg{ii(1),fie})>0

Polyg{fie}=polyg{ii(1),fie};

end

end

ShowGTPolyg(img,Polyg,103);

saveas(103,[outimgdir imagename(1:end-4) '_boxlayout.png']);
```

Figure 5.8 box layout

5.3.1.5 Display Surface layout

Finally display the surface layout of the room by considering box layout and the best candidate layout from above step.

```
hsvmask=rgb2hsv(mask color);

hsvmask(:,:,3)=aa*255;

% hsvmask(:,:,2)=aa;

mask_color=hsv2rgb(hsvmask);

tempimg = double(img)*0.5 + mask_color*0.5;

% tempimg = mask_color;

imshow(uint8(tempimg));

saveas (102,[outimgdir imagename(1:end-4) '_surfacelabels.png']);
```

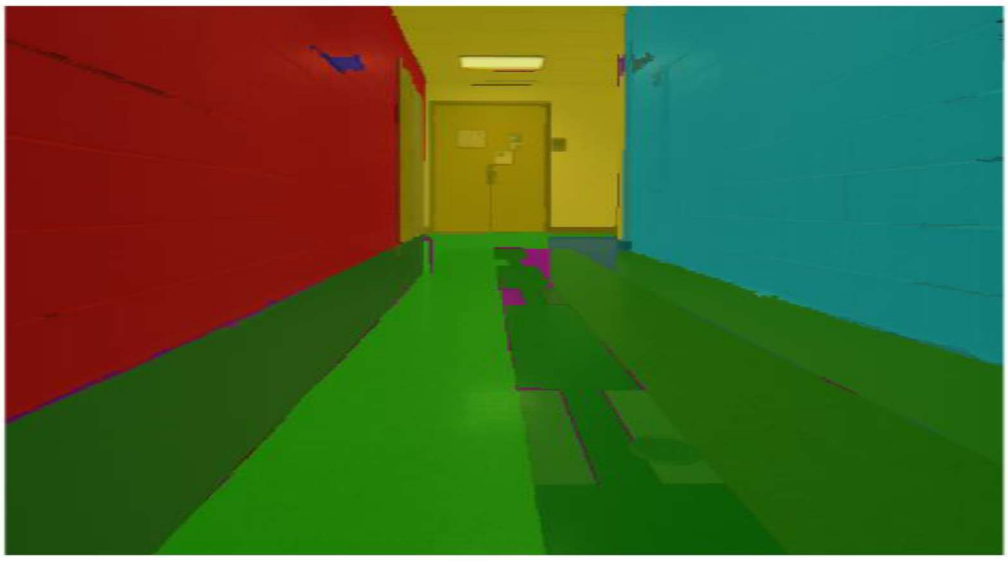


Figure 5.9 surface layout

5.3.2 Aggregate channel features(ACF) implementation

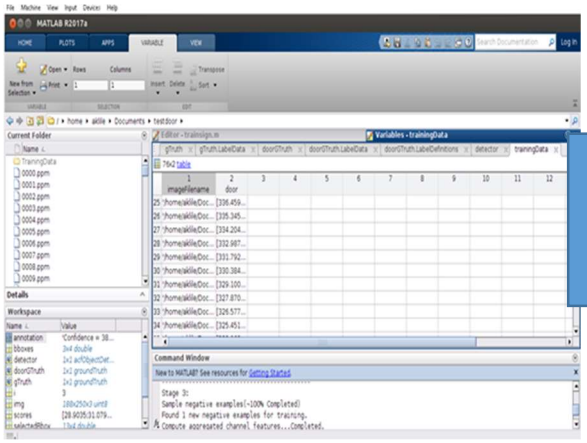
In this section implementation of Aggregate channel features(ACF) is discussed. After room layout recovery algorithm is run the output is used to train Aggregate channel features(ACF) object detector. The output image of room layout recovery algorism is first labeled again before it is used to train ACF detector.

5.3.2.1Train ACF object Detector

By Using the trainACFObjectDetector function with training images it is possible to create an ACF object detector that can detect doors in this case (same for sign and text).

trainingData=

objectDetectorTrainingData= (doorGTruth, …



Training data for door

Train ACF Detector

ACF door detect

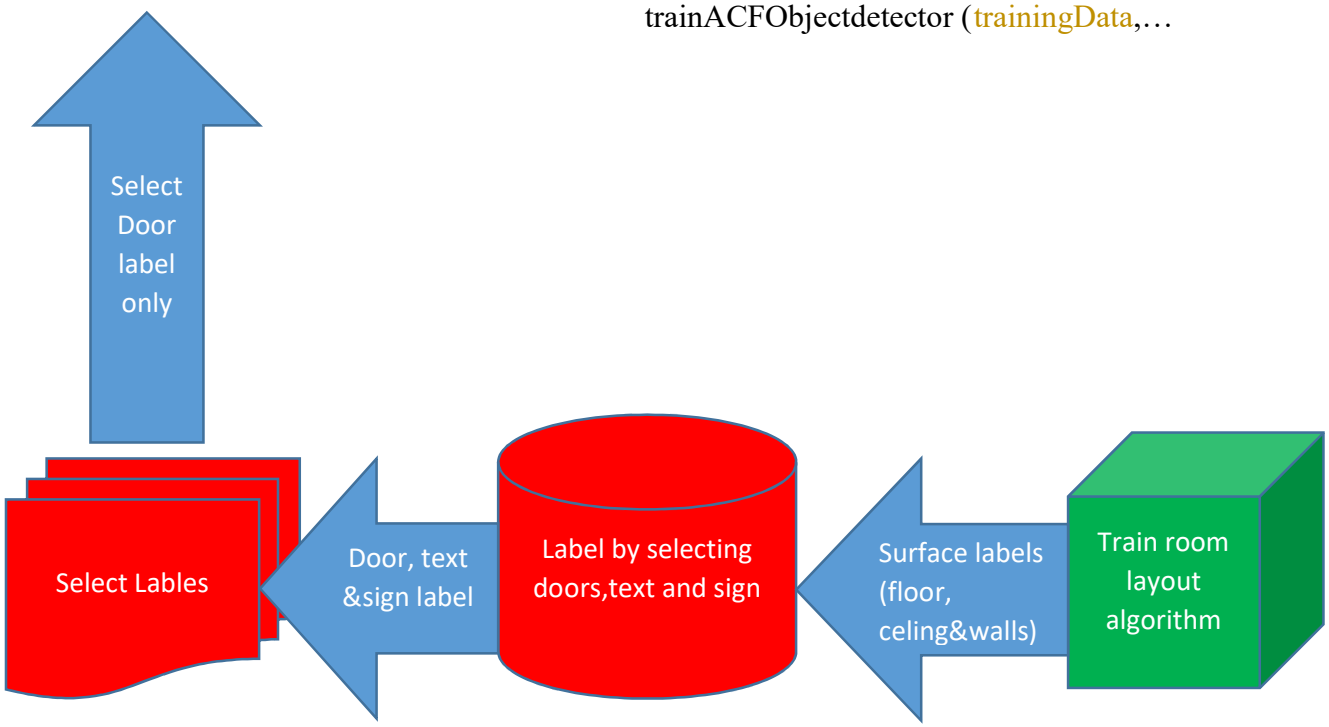
ACF Object Detector

[bboxes,scores]=

detect (detector…)

detector=

trainACFObjectdetector (trainingData,…



doorGTruth=

gtruth

surface labels

splitLables (gtruth,..)

Figure5.10 ACF Training procedure for door detection (the same procedure is used for text and sign detection)

5.3.2.1.1First Load training Data

Load the gTruth object from a saved MAT file. The ground truth is the set of known locations of objects of interest in a set of images to be used to train the detector

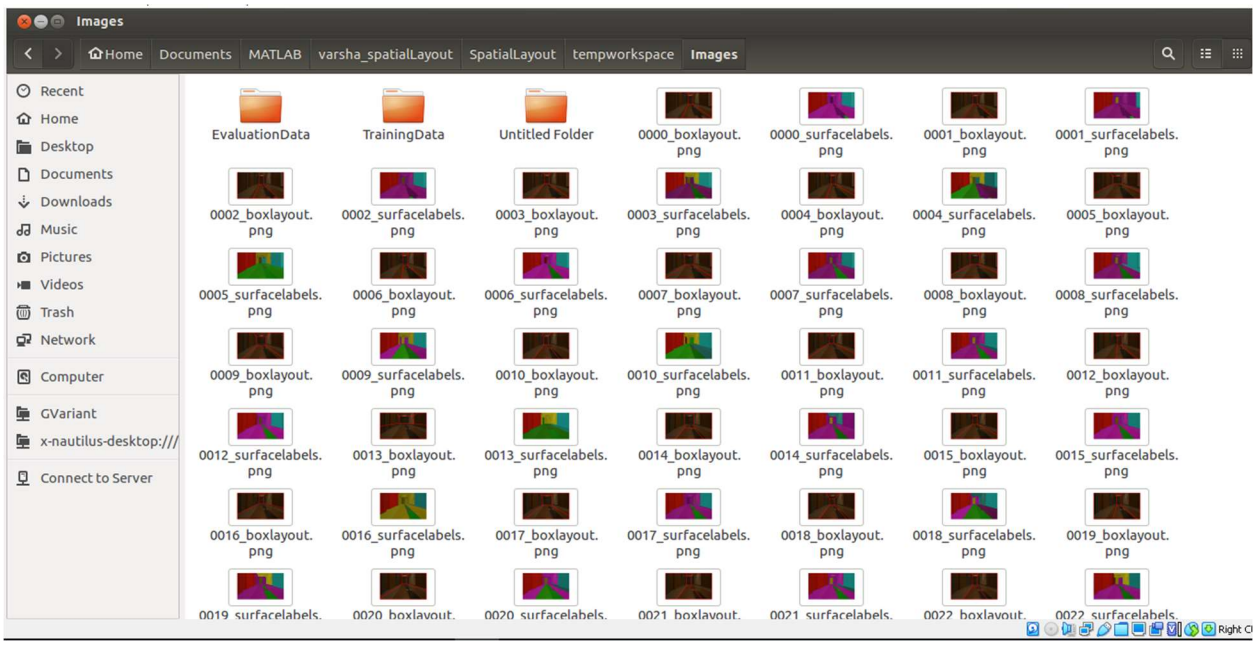


Figure 5.11 ACF Ground Truth Dataset

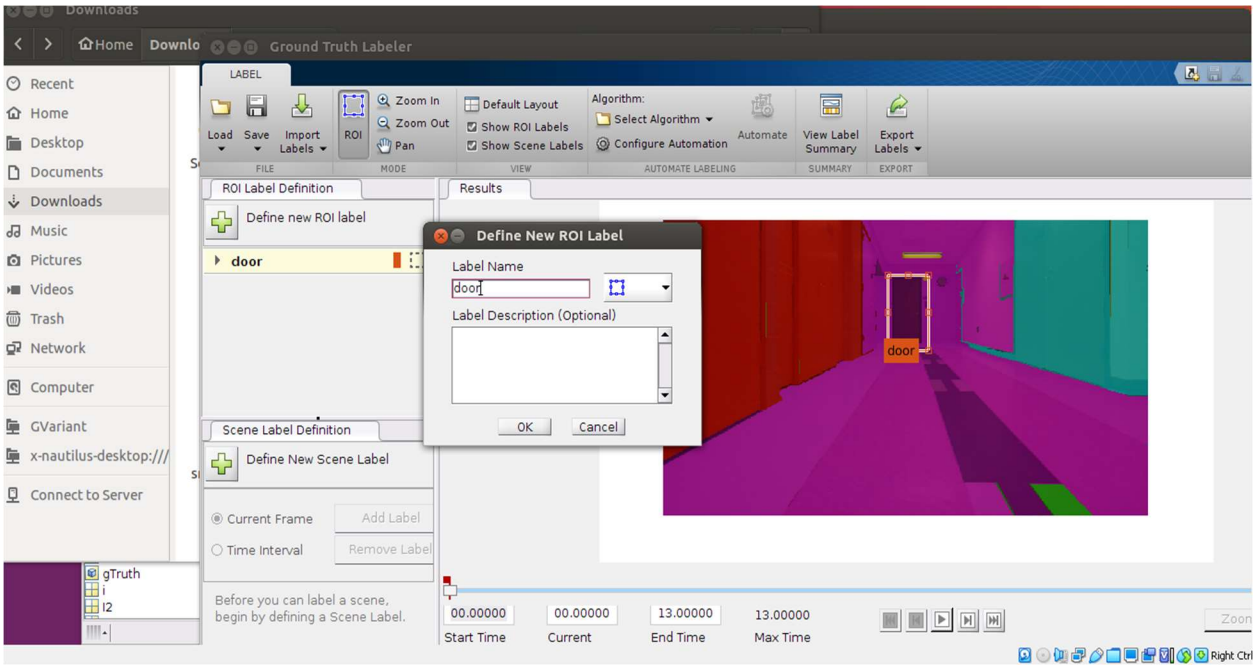


Figure 5.12 Label Ground Truth Data

load('gTruthsegemented2816final.mat');

// Create Training Data from Ground Truth

Isolate ground truth data by their association with the 'door label' label. This creates a new gTruth object that contains the ground truth of only door

doorGTruth = selectLabels (gTruth,'door');

workspcdir='/home/aklile/Documents/MATLAB/varsha_spatialLayout/SpatialLayout/tempworkspace/Images/TrainingData';

if ~exist(workspcdir,'dir')

mkdir(workspcdir);

end

addpath('workspcdir');

5.3.2.1.2 Extract a subset of the Ground Truth dataset

TrainingData is a table that contains training data from the ground truth.

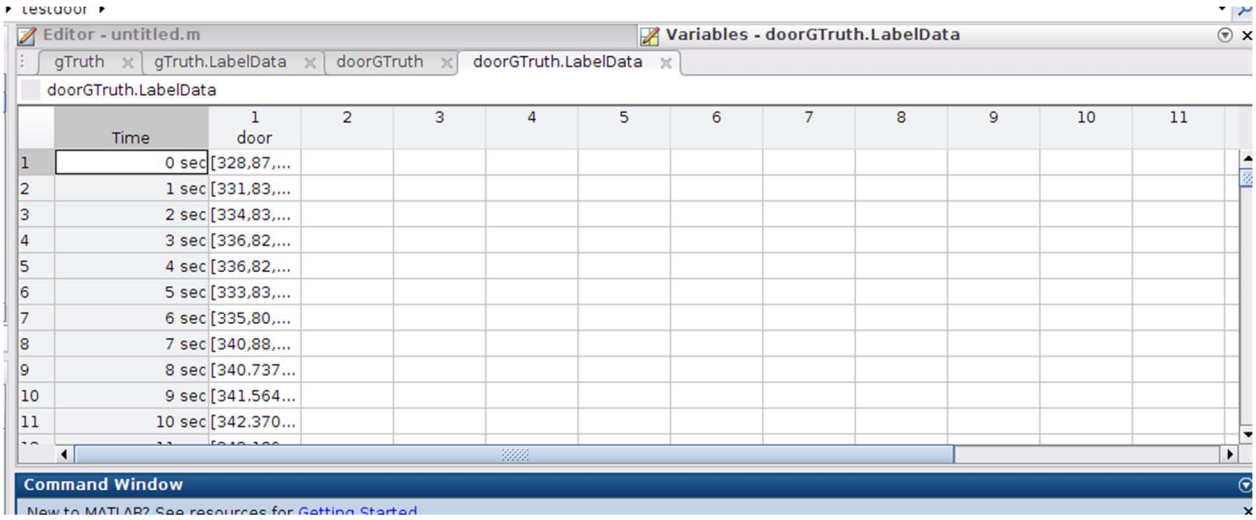


Figure 5.13 Split labels

trainingData = objectDetectorTrainingData(doorGTruth,'SamplingFactor',9,...

'WriteLocation','TrainingData');

5.3.2.1.3Train the ACF Detector

detector = trainACFObjectDetector(trainingData,'NumStages',3,'ObjectTrainingSize',[100 100]);

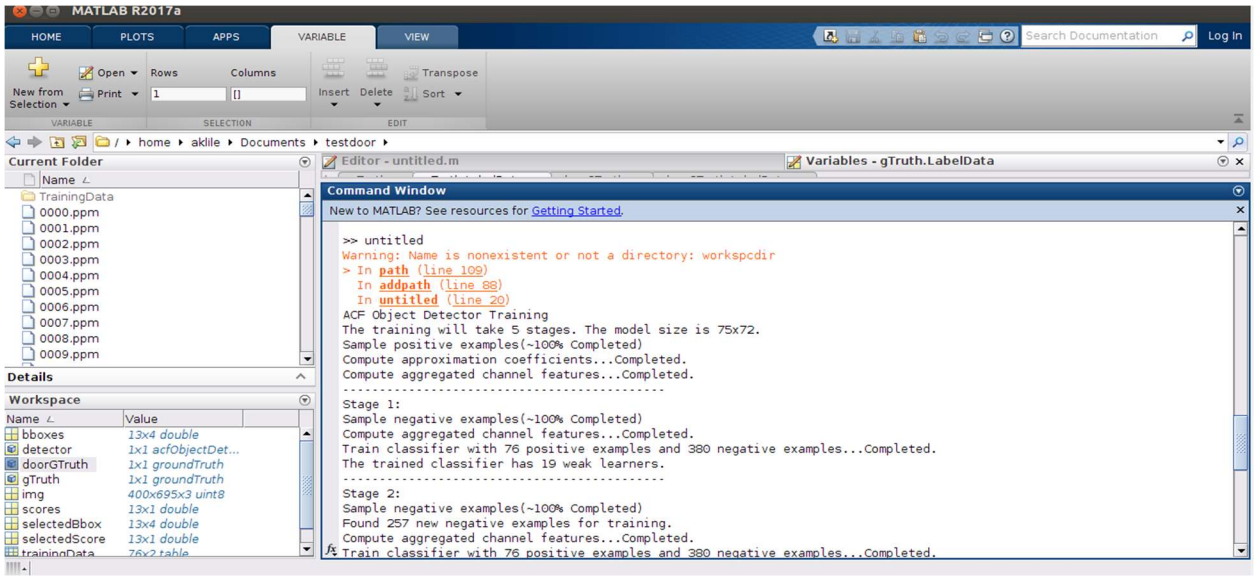


Figure 5.14 train ACF detector

%Save the detector to, a MAT file,'ObjectTrainingSize',[150 150]

save('Detectorsegmentedfull2816707.mat','detector');

rmpath('TrainingData');

5.3.2.1.4 check detection

img = imread('test2.ppm');

[bboxes,scores] = detect(detector,img);

for i = 1:length(scores)

annotation = sprintf('Confidence = %.1f',scores(i));

img = insertObjectAnnotation(img,'rectangle',bboxes(i,:),annotation);

end

figure

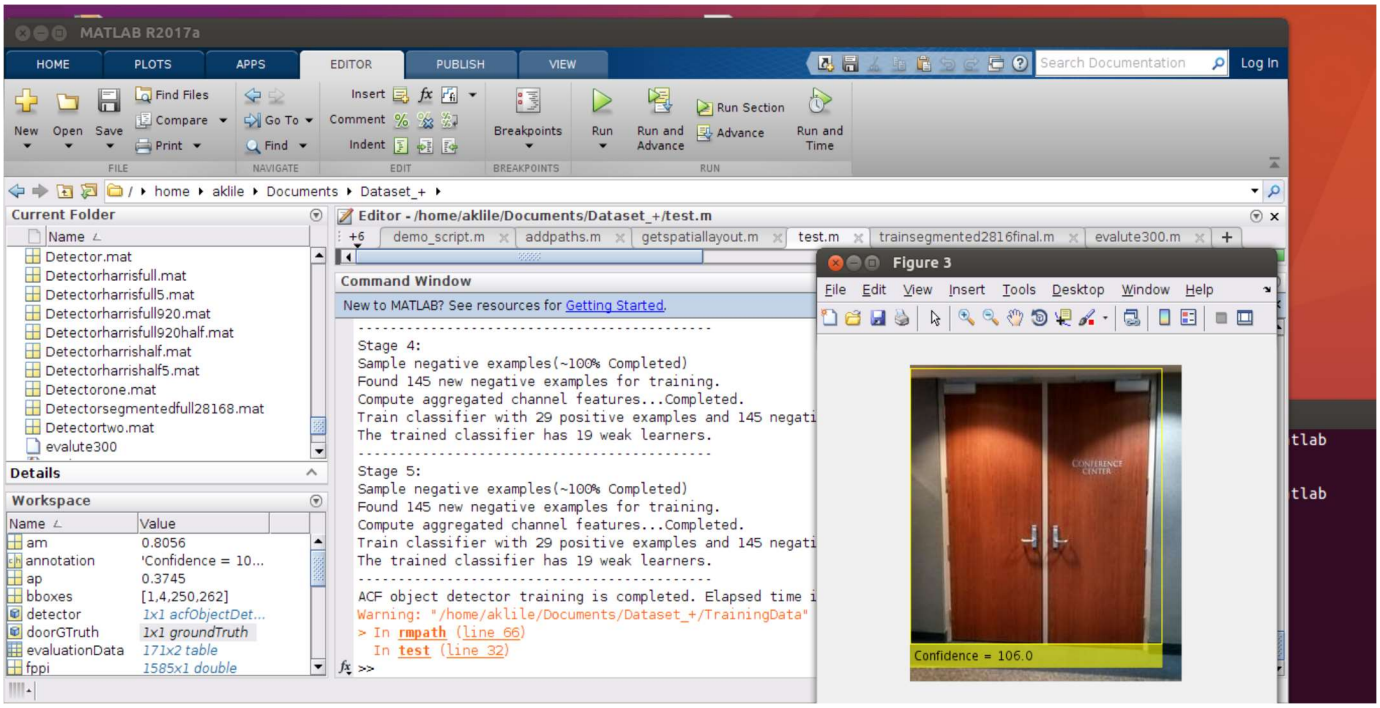


Figure 5.15 sample result after ACF detection

5.3.2.2 Evaluate and test ACF object detector

After training the ACF object detector the next step is testing and evaluating the performance of the detector with separate detest.

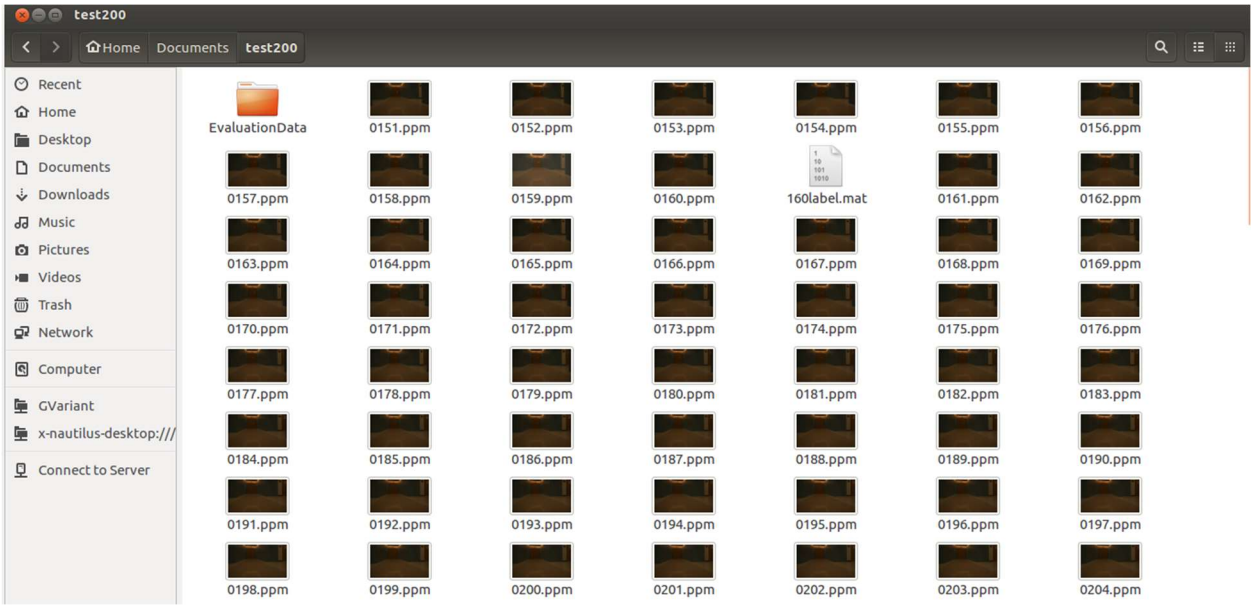


Figure5.16 testing data set

5.3.2.2.1Load Ground Truth Data for Testing

Create a separate and different ground truth data set that is not used for training stage

```
load('gTruthEvaluation.mat')
```

5.3.2.3Create Dataset from Testing Ground Truth

Create a subset of ground truth with label door test.

```
doorTestGTruth = select Labels(gTruth,'doorTest');
```

5.3.2.4 Evaluate Detector

```
% Evaluate Metrics

[ap,recall,precision] = evaluateDetectionPrecision(results...

    ,evaluationData(:,3),overlap);

[am,fppi,missRate] = evaluateDetectionMissRate(results,evaluationData(:,3),overlap);

% Plot Metrics

subplot(1,2,1);

plot(recall,precision);

xlabel('Recall');

ylabel('Precision');

title(sprintf('Average Precision = %.1f', ap))

grid on
```

5.4 Mapping implementation

Since ROS developed in unit of nodes each response or decision the robot has to make have to be programmed separately as node. Finally, each node makes communication with each other to make the whole system work as required. So a master that manages connection information in a message communication between nodes is an essential element that must be run first in order to use ROS. Each node described in the above is programmed as node and configured the nodes as subscriber or publisher node in CMakeList.txt and Message.txt folder for the type of nodes and the message they used to exchange respectively in workspace package. After building the packages and launching each nodes, each nodes begun to communicate with the master and create the whole system that operate the navigation, path planning, obstacle avoidance, sign recognition, controller the action of the robot based on the sign, current position estimate etc. through communicating and exchanging information with their required node until termination of the whole system.

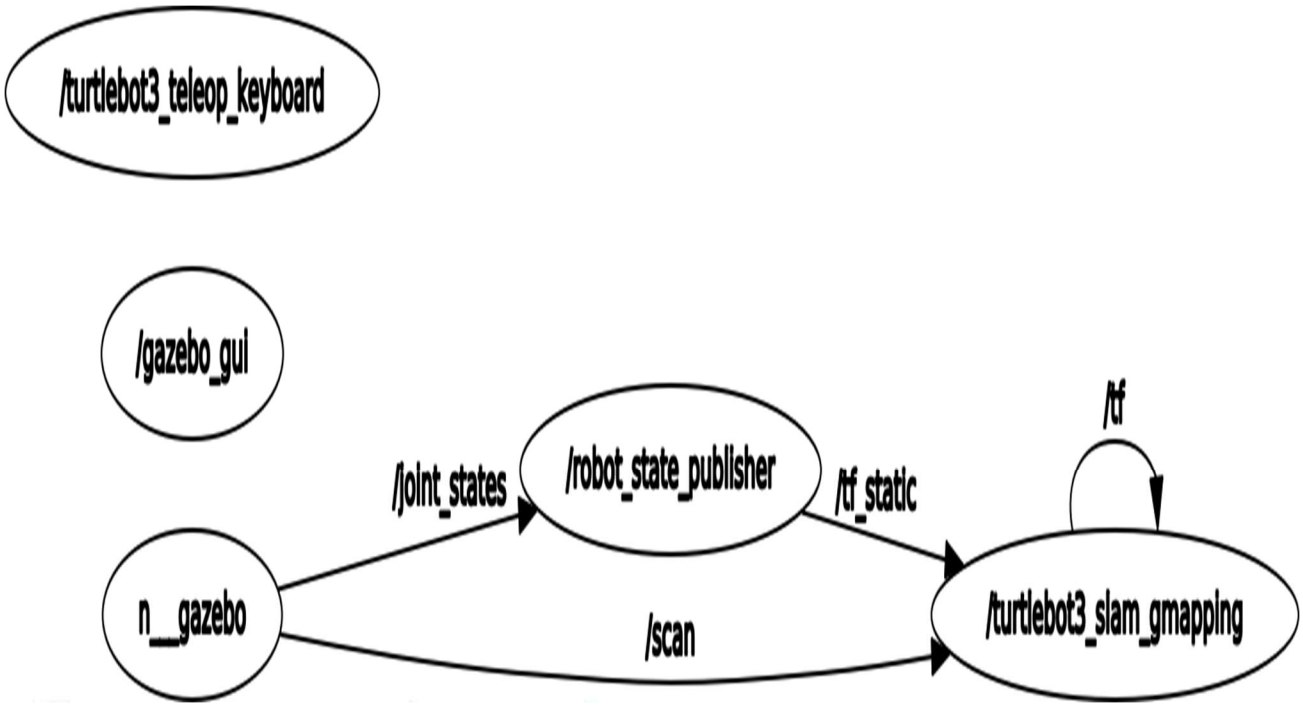


Figure 5.17 Mapping Graph Created by Rqt that show the connection and exchange of message between subscriber and publisher node with their respective types of topic

CHAPTER SIX

6. Result, Evaluation and Discussion

6.1 Overview

To test the effectiveness of the proposed scene analysis plus aggregate channel features (ACF) method First experiment was conducted to choose the best feature extractor algorithm appropriate for the method. Second to find the optimal number of training size the effect of the Number of training images are determined. Finally, the effect of integrating scene analysis with aggregate channel features (ACF) is shown.

6.2 Experiment using Aggregate Channel features (ACF)

To check the effectiveness of the proposed integration of room layout recovery and Aggregate Channel features (ACF) object detector it is necessary to measure the effect of only using Aggregate Channel features (ACF) for object detection and compare the performance when room layout recovery algorithm is added.

Before proceeding with evaluation of the performance of the two approaches first selecting the appropriate feature extractor for training the algorithms must be answered.

6.2.1Performance of different feature extraction algorithm

Here I considered the effect of using different feature extractors on the precision and number of false positive per image to be matched. For this experiment 130 training images are used and trained with aggregate channel features without incorporating room layout recovery algorithm. The 130 training images are labeled and three feature extractors that are available in ground truth labeler application of Mat lab are used. These are Minimum Eigen value feature extractor, SURF feature extractor and Harris feature extractor. The results are shown below. For testing 50 images are used

With scaled down image, as one can see from images below, minimum Eigen value have less false positive rate than the other algorithm. On the other hand, Harris detector has the same precision as minimum Eigen value. and finally surf has zero average precision. It is clear that minimum Eigen value is slightly better than Harris detector and is more reliable for matching between trainee and test images

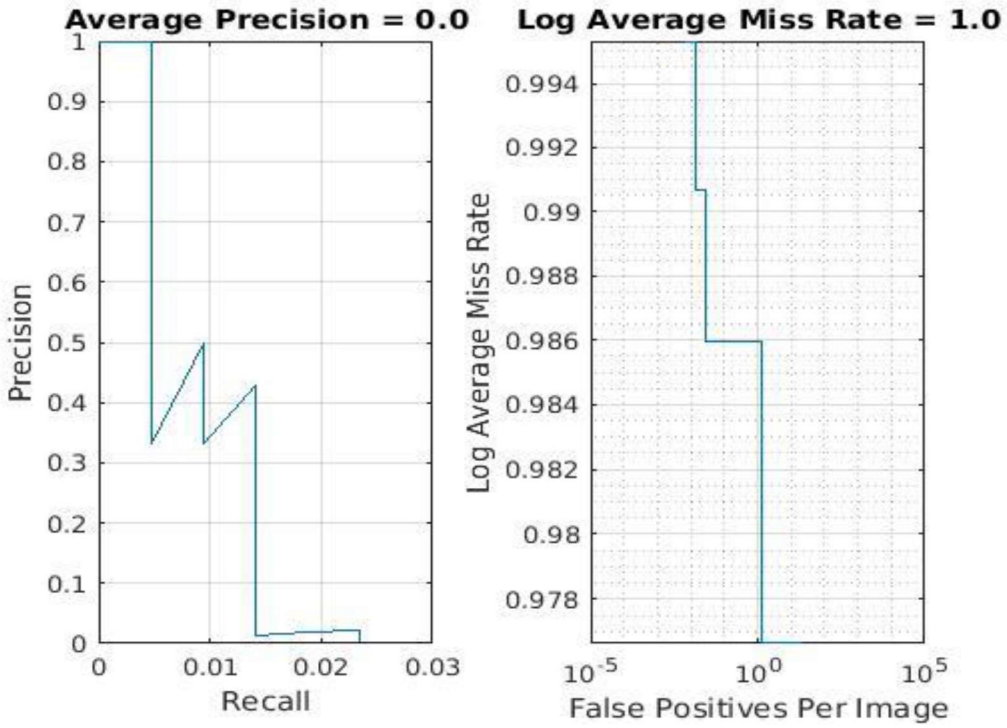


Figure 6.1 detector surf

Surf detector registered average precision of zero and log average miss rate of 1.0 which is the worst performance compared to the other two feature extractors.

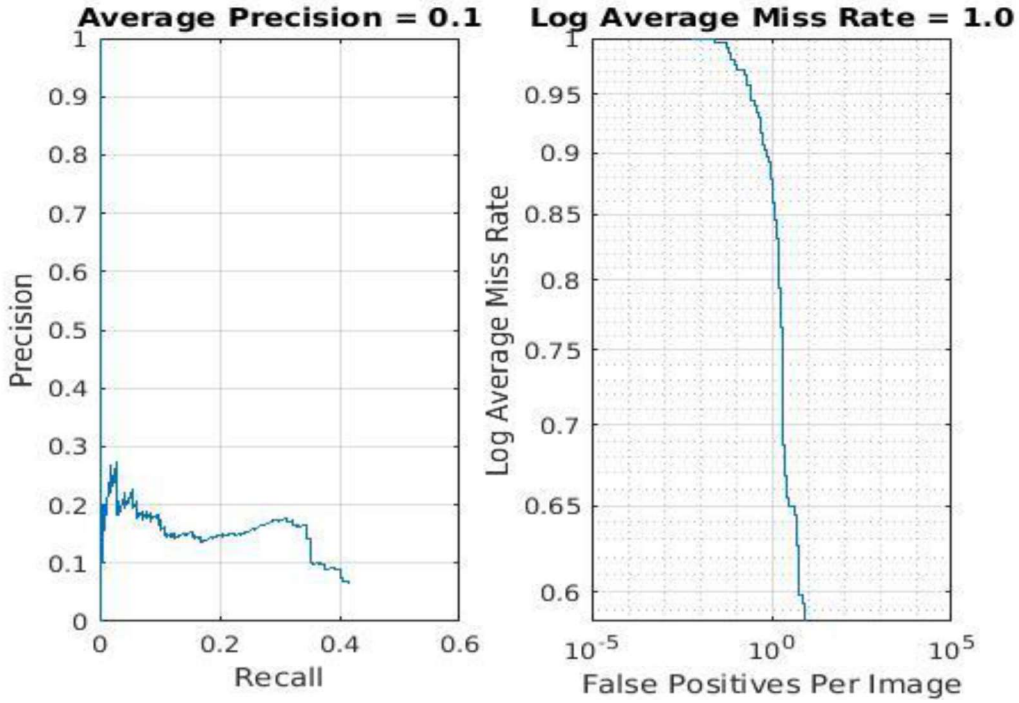


Figure 6.2 Harris detector

Harris detector have a slightly better average precision than surf when it is trained with 130 images

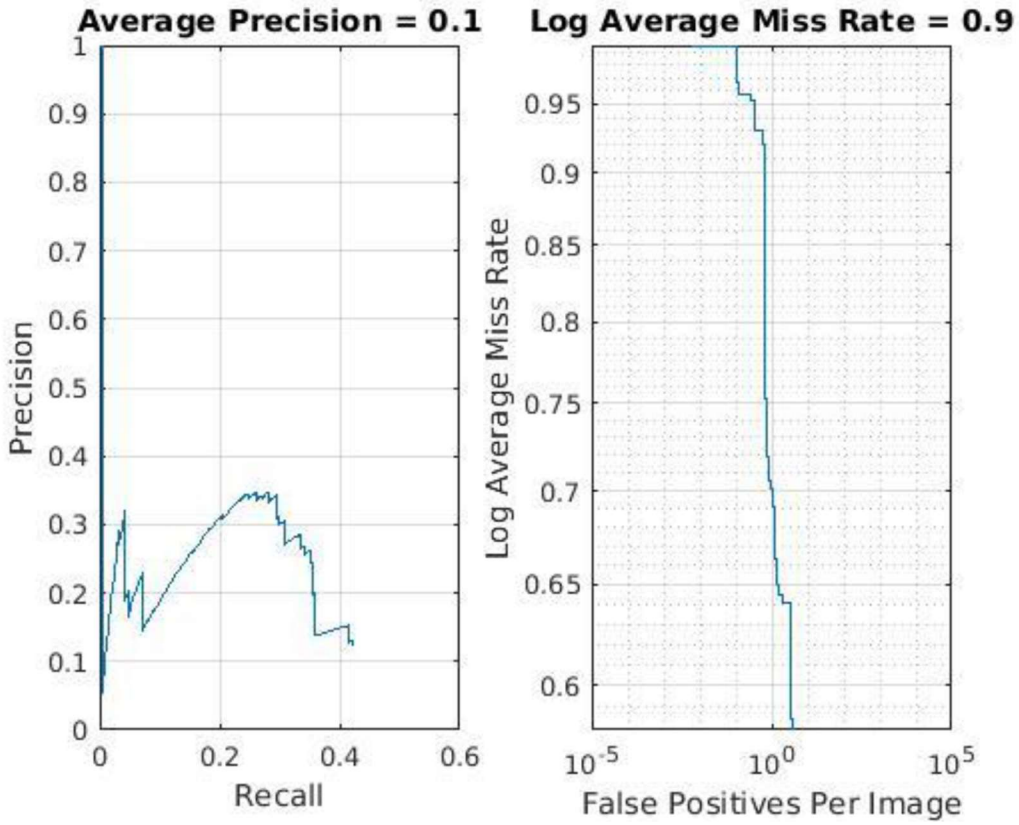


Figure 6.3 Detector Minimum Eigen value

Minimum Eigen value not only improves precision but also register a slight decrease in the log average miss rate.

6.2.2 Effect of size of training image on performance

In this case the effect of increasing the size of training image on the precision and number of false positive per image to be matched is considered. Moreover, rather than merely increasing the number of images for training without improving performance of the algorithm determining the optimal training image size advantageous. The results are given below. In this case I use the same feature extractor namely Harris detector with different training image size.one can see from the figure below that increasing the number of training images improve the performance of the algorithm. The number of Testing image is 50.

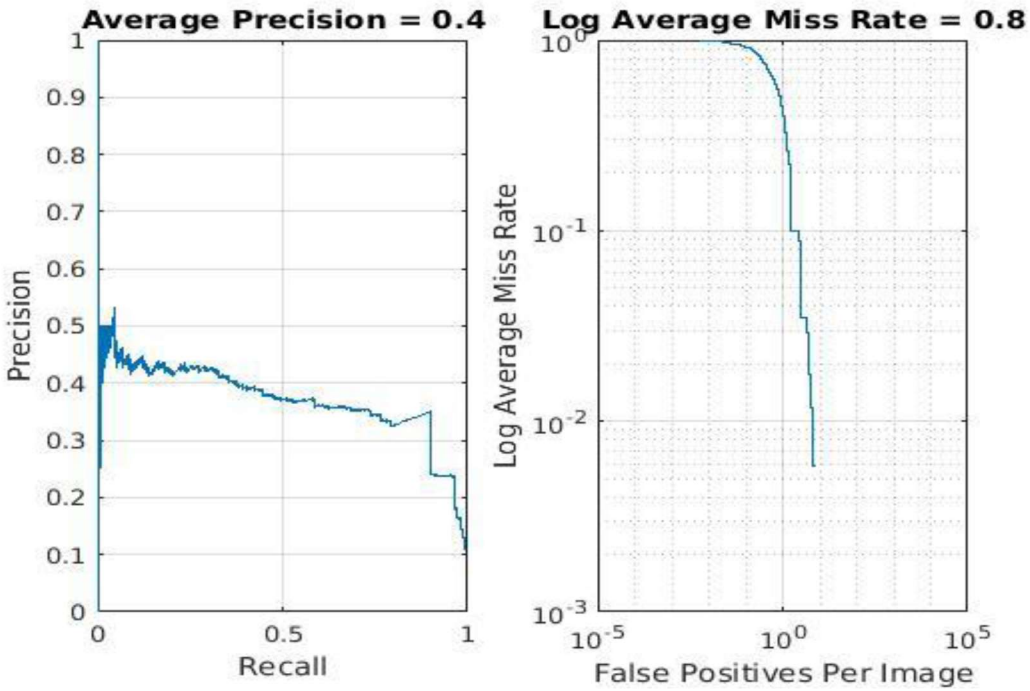


Figure 6.4 Harris detector trained with ACF (170 images)

By just adding 40 images on 130 images compared to figure 6.2 the precision increased from0.1 to 0.4 while log average miss rate decreased to 0.8

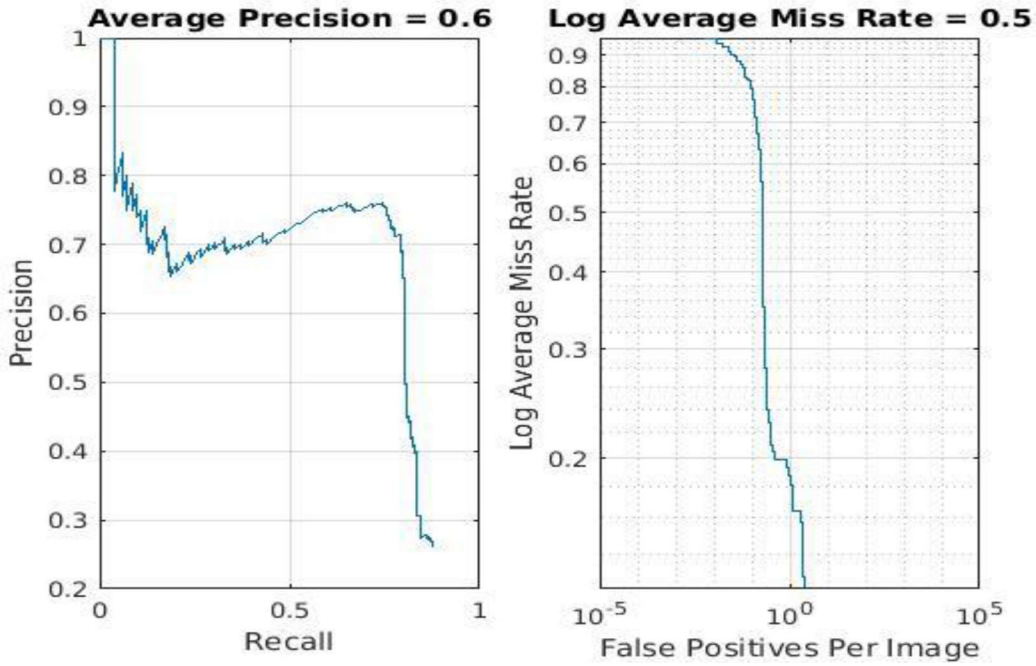


Figure 6.5 Harris detector trained with ACF (300 images)

Increasing the training image to 300 increased the precision to0.6 while decreasing the log average miss rate to 0.5

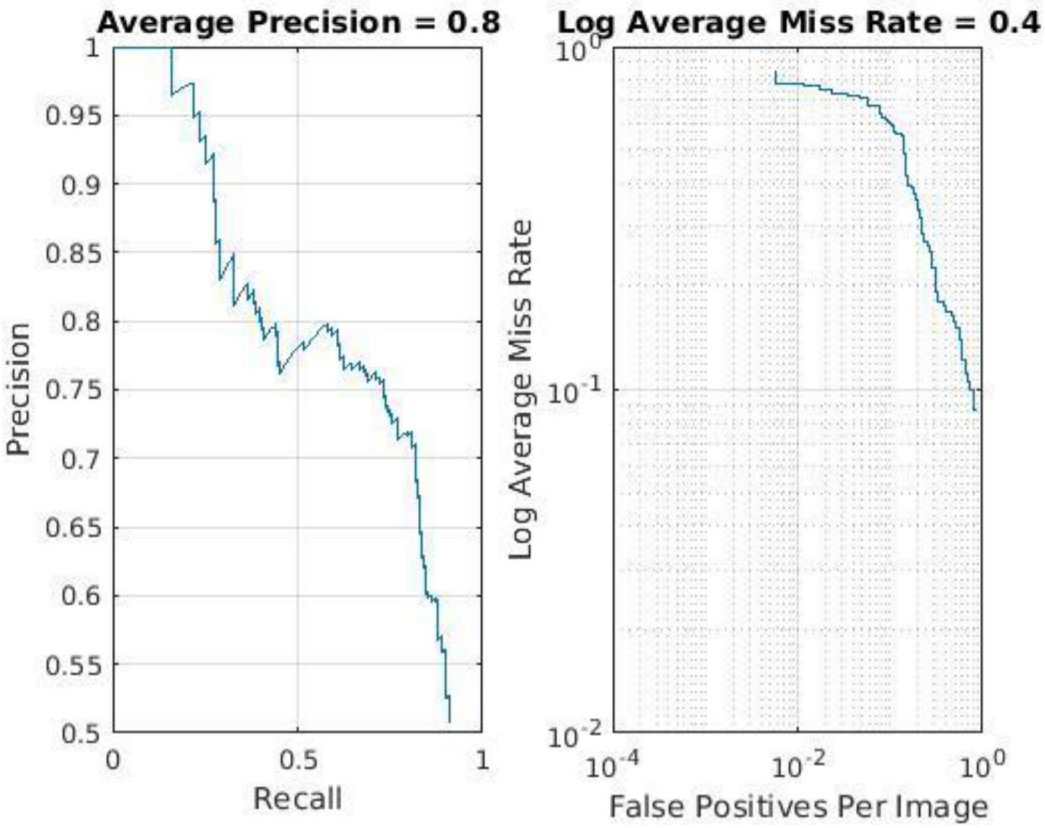


Figure 6.6 Harris detector trained with ACF (900 images)

Training with 900 images using Harris detector give as an average precision of 0.8 and log average miss rate of 0.4 which is a significant improvement.

6.3Experiment integrating room layout recovery with Aggregate Channel Features (ACF)

For this section the effect of integrating room layout recovery algorithm with Aggregate Channel Features (ACF) is discussed.

6.3.1 Effect of scene analysis

For this case first room layout recovery algorithm is trained and the output, which is geometric classes (walls, ceiling and floor) labeled with different colors, is used to train ACF object detector instead of raw image the effect is discussed below.

6.3.1.1 Door detection

First I consider the problem of door detection. As explained above room layout recovery and ACF object detector is used together. To investigate the problem further, I consider when the door is fully visible and when the door is only partially visible

6.3.1.1.1Door detection when door is fully visible

In this case the effect of scene analysis on the precision and number of false positive per image to be matched is considered. First scene analysis method is used to recover the spatial layout and recover free space. Then the output is used to train aggregate channel features (ACF) detector. The results are given below as we can see from the figure the average precision is improved significantly while false positive rate also dropped considerably. For this experiment 900 training images are used and trained with scene analysis method first and incorporating aggregate channel features later. The number of test image is 300.

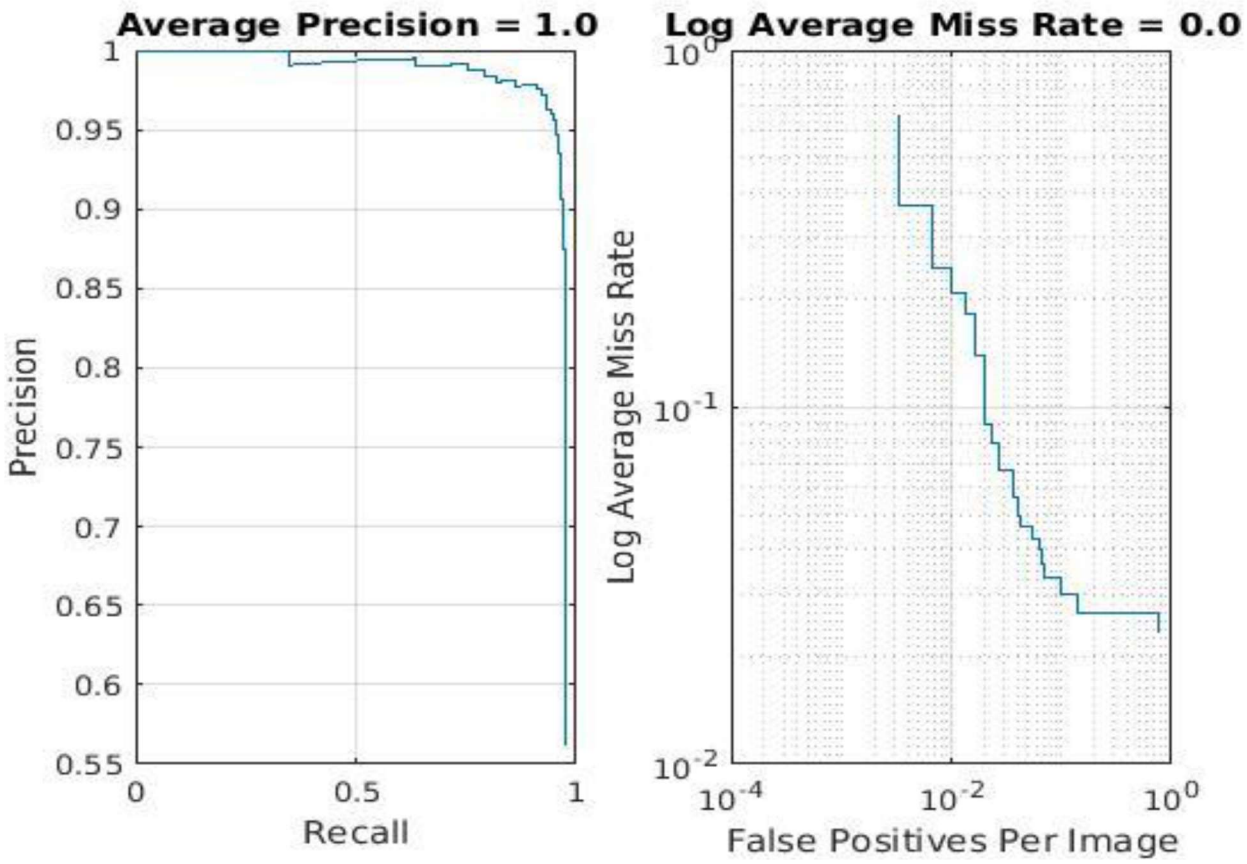


Figure 6. 7 Harris detector trained with ACF +SCENE ANALYSIS (room layout recovery) (900 images)

When ACF object detector is trained on the output of room layout recovery algorithm while training image size is constant and the same feature extractor is used as in figure 6.6 the performance improved significantly. Average precision of 1 is achieved while Log Average Miss Rate is reduced to 0.0. proving that the proposed integration of ACF object detector with room layout recovery algorithm enhance the performance of the algorithms.

Some results are shown below

Figure 6.8 Sample result ACF +SCENE ANALYSIS door detection

Figure 6.9 Sample result ACF +SCENE ANALYSIS door detection

6.3.1.1.2 Door detection when door is partially visible

The performance the algorithm slightly drops when the doors only partially visible. But it is better than algorithms that relay on geometric properties of doors. because such algorithms fail when the door is not fully visible

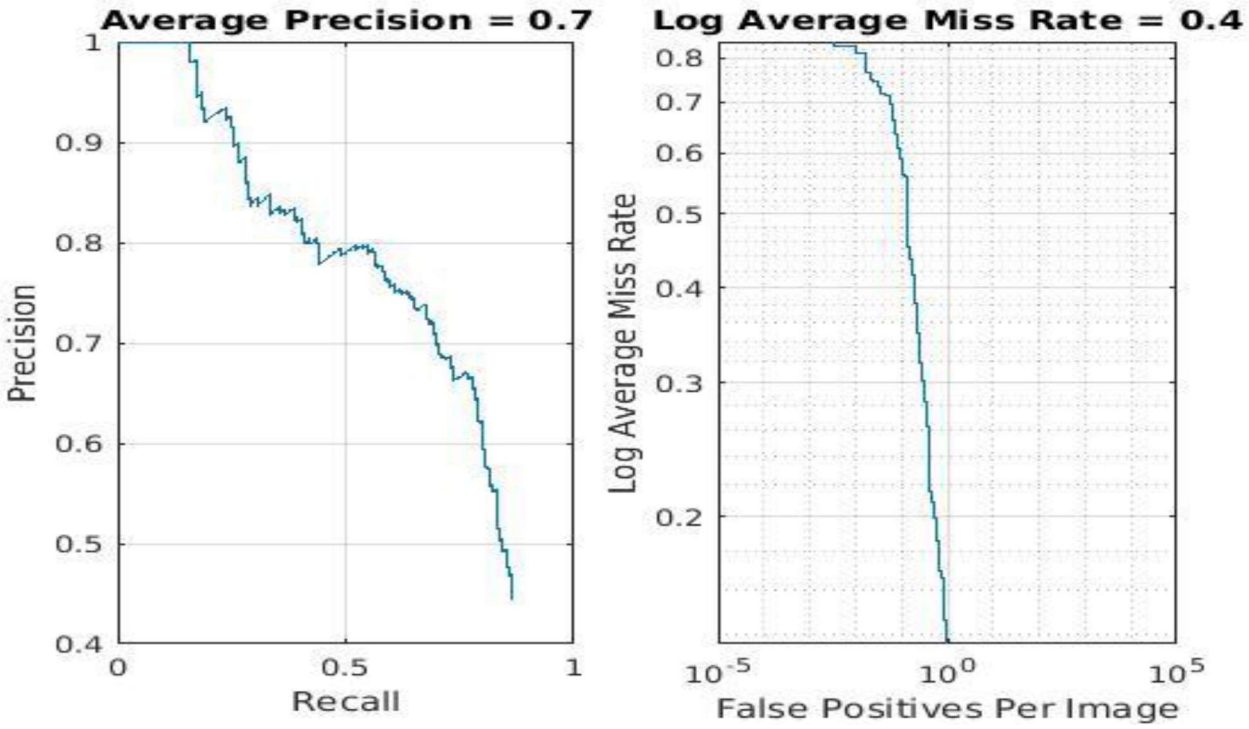


Figure 6.10 Harris detector trained with ACF +SCENE ANALYSIS (room layout recovery) (900 images)

Detecting doors that are partially visible challenging task algorithms that rely on geometric properties totally fail when door is partially visible. In my case the average precision when the door is partially visible is 0.7 while log average miss rate is 0.4.

6.3.1.2 Sign and text detection

The proposed algorithm can detect also detect text and sign as well. The result is shown below. The algorithm is trained with 253 sign images and 247 text images together and tested with 150 images of text and sign.

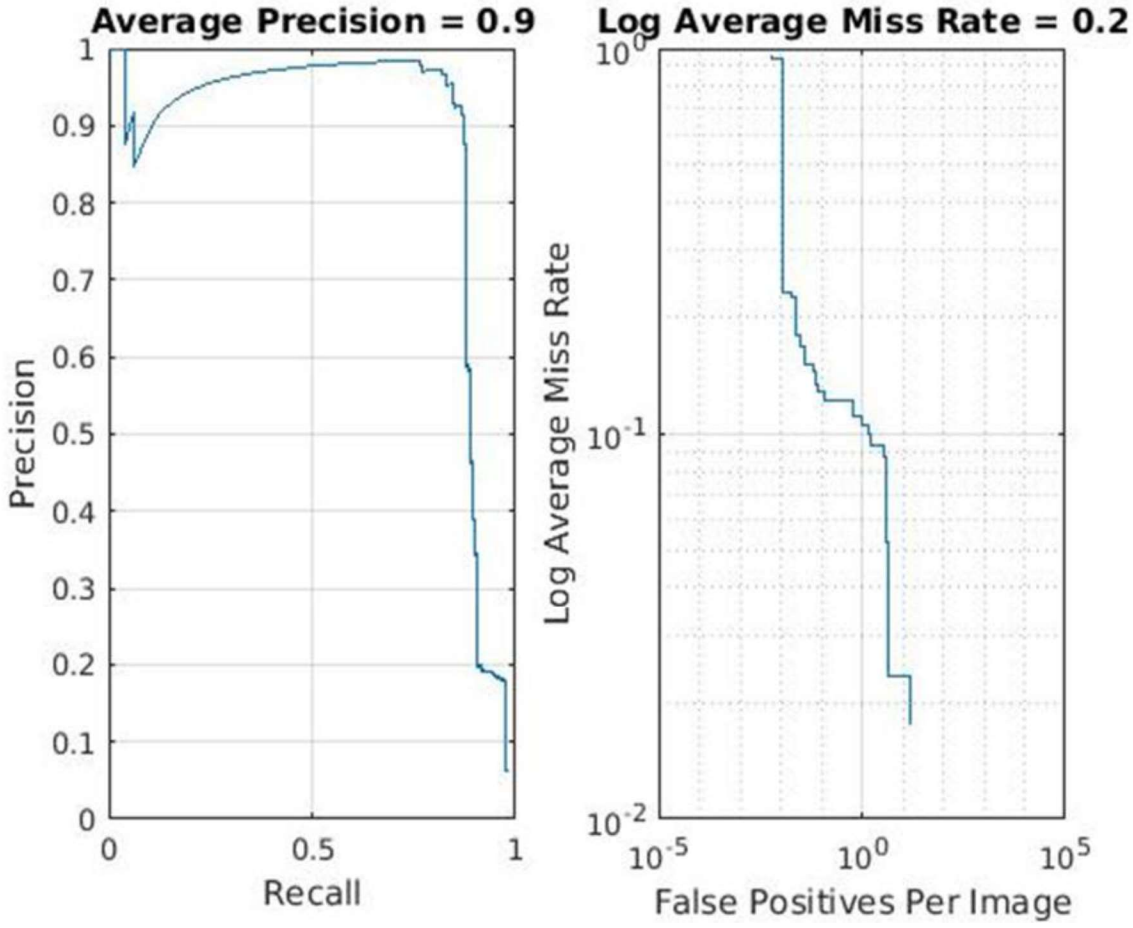


Figure 6.11 Harris detector trained with ACF +SCENE ANALYSIS (*room layout recovery) (500 images)*

Using 500 images of text and sign it is possible to achieve average precision of 0.9 while registering log average miss rate of 0.2

Example of sign detection

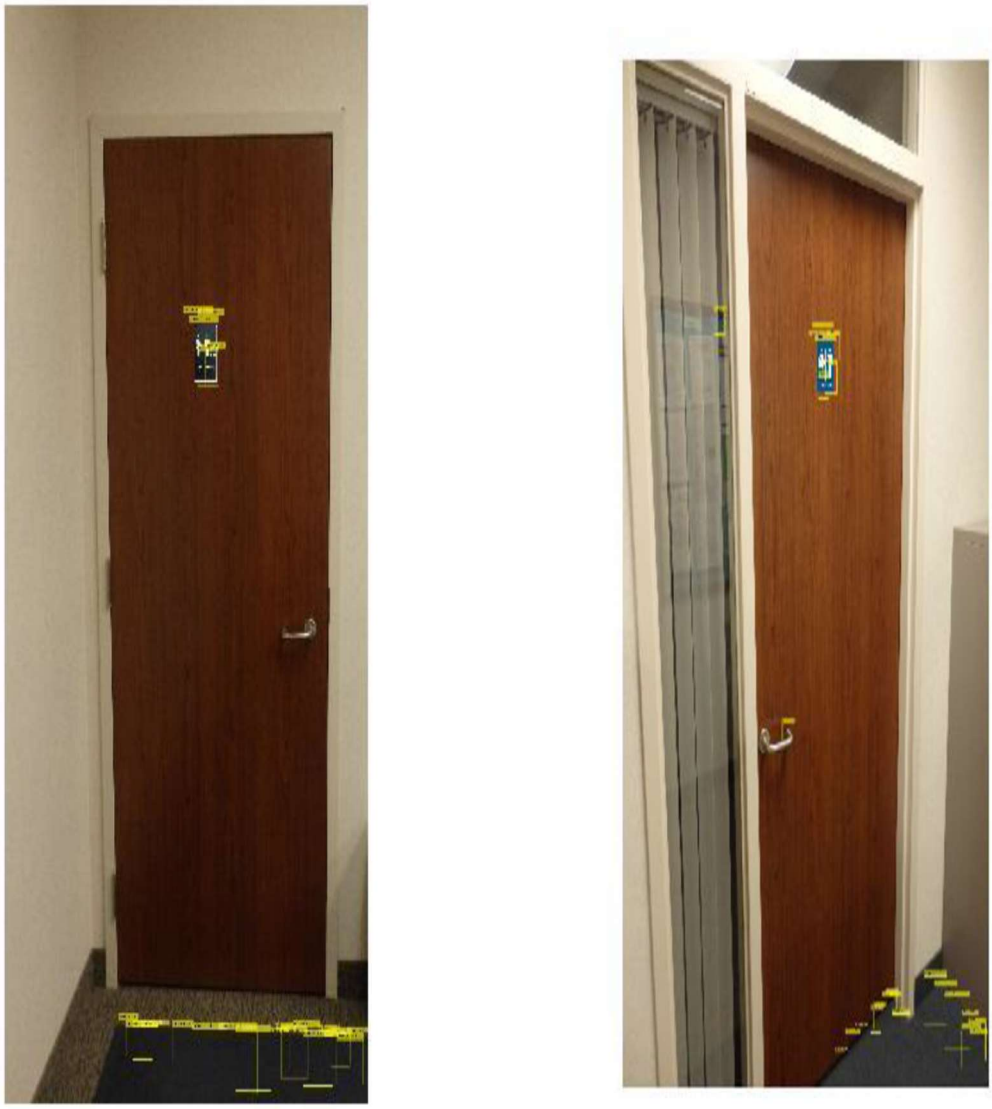


Figure 6.12 sign detection

Example of text detection

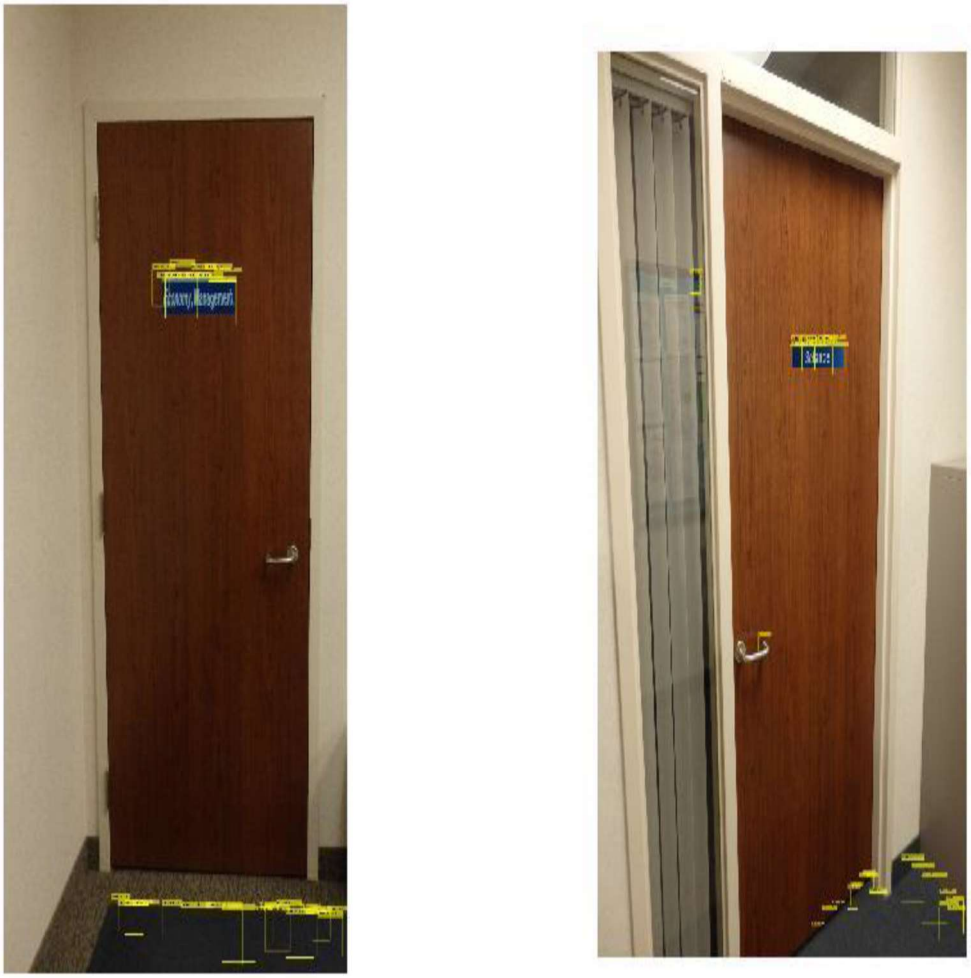


Figure 6.13 text detection

6.4 Scene analysis for Mobile Robot Navigation

Since robot navigation focus on generating a global path to navigate the robot from source to destination it need to create 2D geometric representation of its environment which is called map using the data provided from the LIDAR sensor. So using the method SLAM and Gmapping package provided by ROS a 2D base map was created by using LIDAR laser and odometry data. Using model of the indoor environment built and turtle bot3 simulation the LIDAR sensor and

odometry, localization have been collected and build the map on the riviz simulator which used by the robot for navigation.

6.4.1 Environment Mapping

The map of the environment created by using SLAM The mapping process is visualized using the ROS visualization tool (RViz). The image then shown as a two-dimensional Occupancy Grid Map (OGM) created at an early stage and a green line that represents the scan information from the LiDAR. The white represents the area that the robot can move and the black represents the occupied area

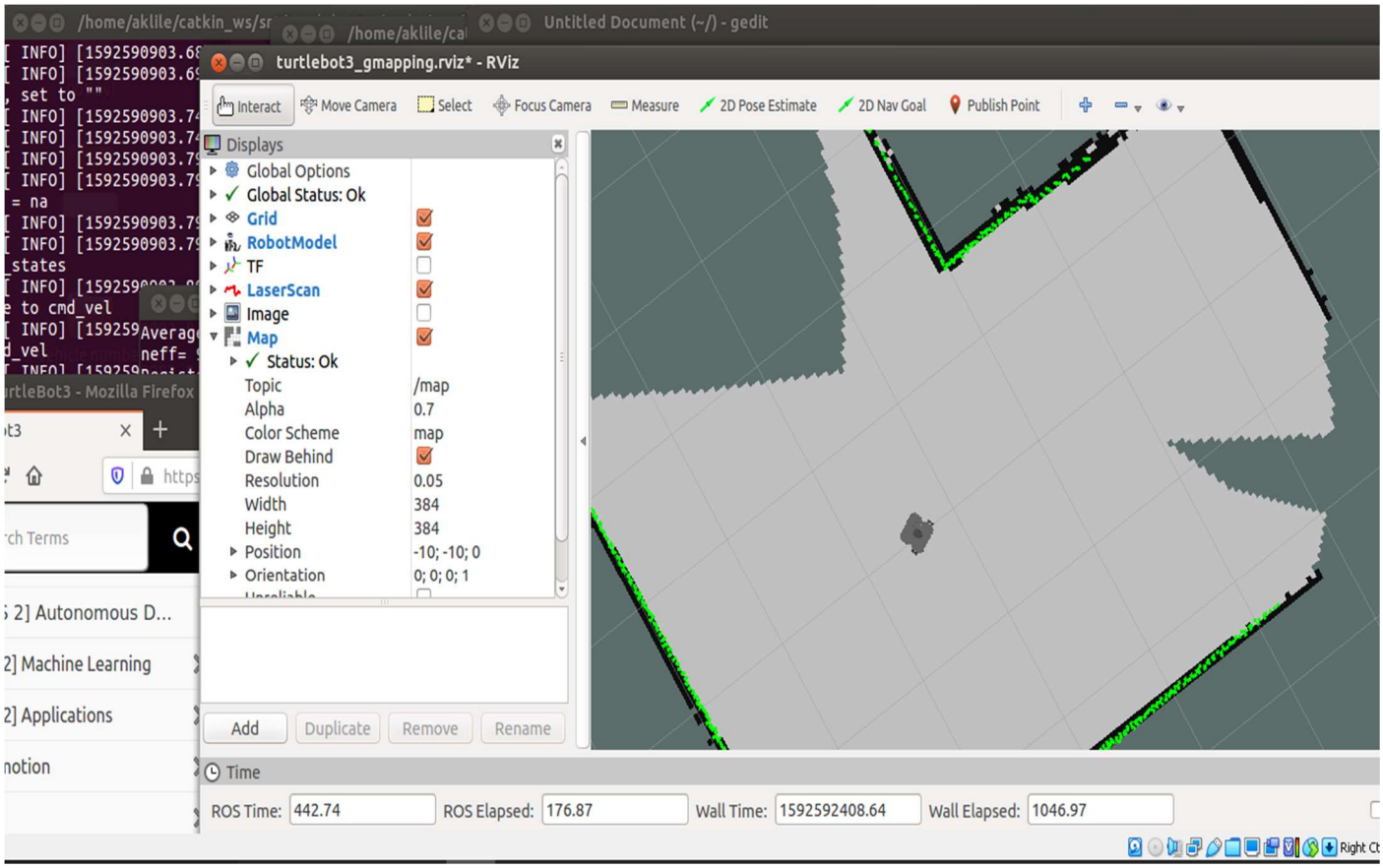


Figure 6.14 The Mapping Process Sample Visualized by RViz and RQT

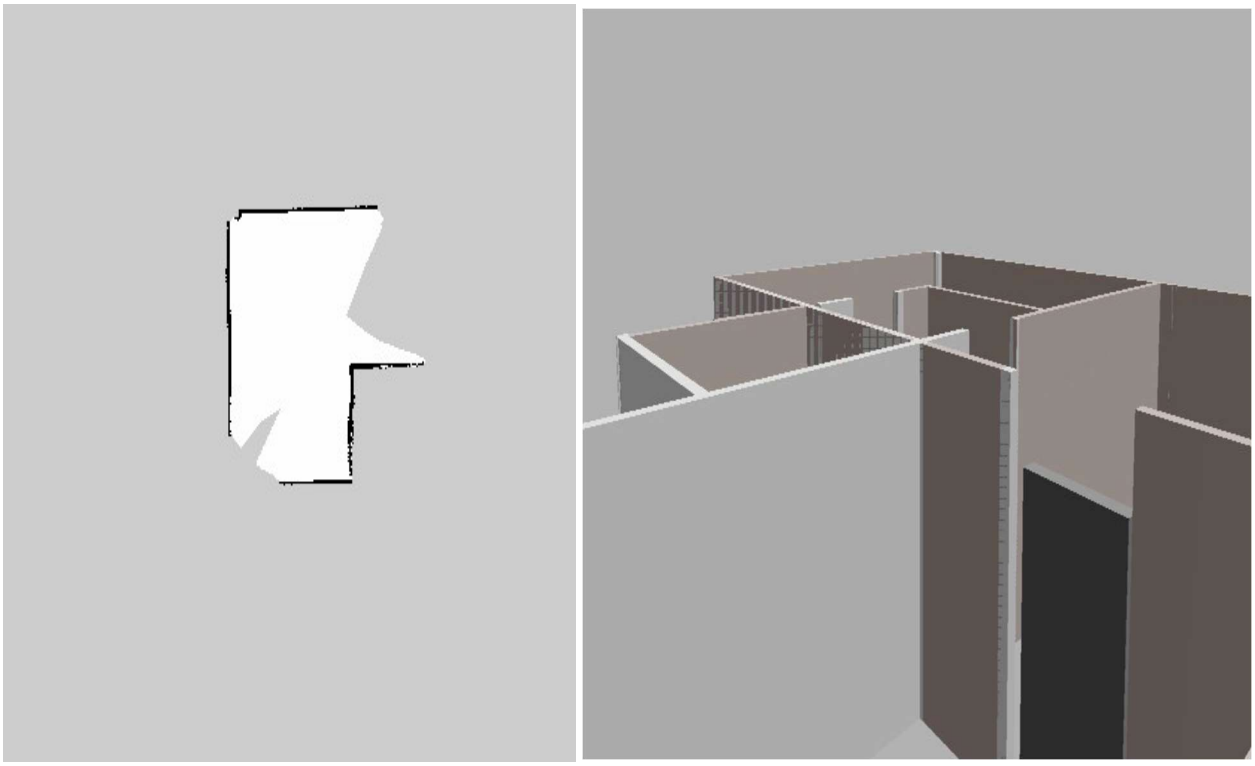


Figure 6.15 Partial Map Created for model indoor environment

The above figure shows the created map and the model world file created.

6.4.2 Navigation

Using the created base environment and map in the mapping step, the Turtlebot can navigate the environment from some point to another The scenario described below is to navigate the environment from the starting room 1 to the door of the room. First, the Turtlebot should know where the door is and should be able to navigate to that location and look for the door. Room layout recovery algorithm and aggregate channel features (ACF) detector are used together to detect the door at some location when building a map and then remember that location for the future. When the robot is given a task to find that door, it just has to navigate to that location and start looking for it again.

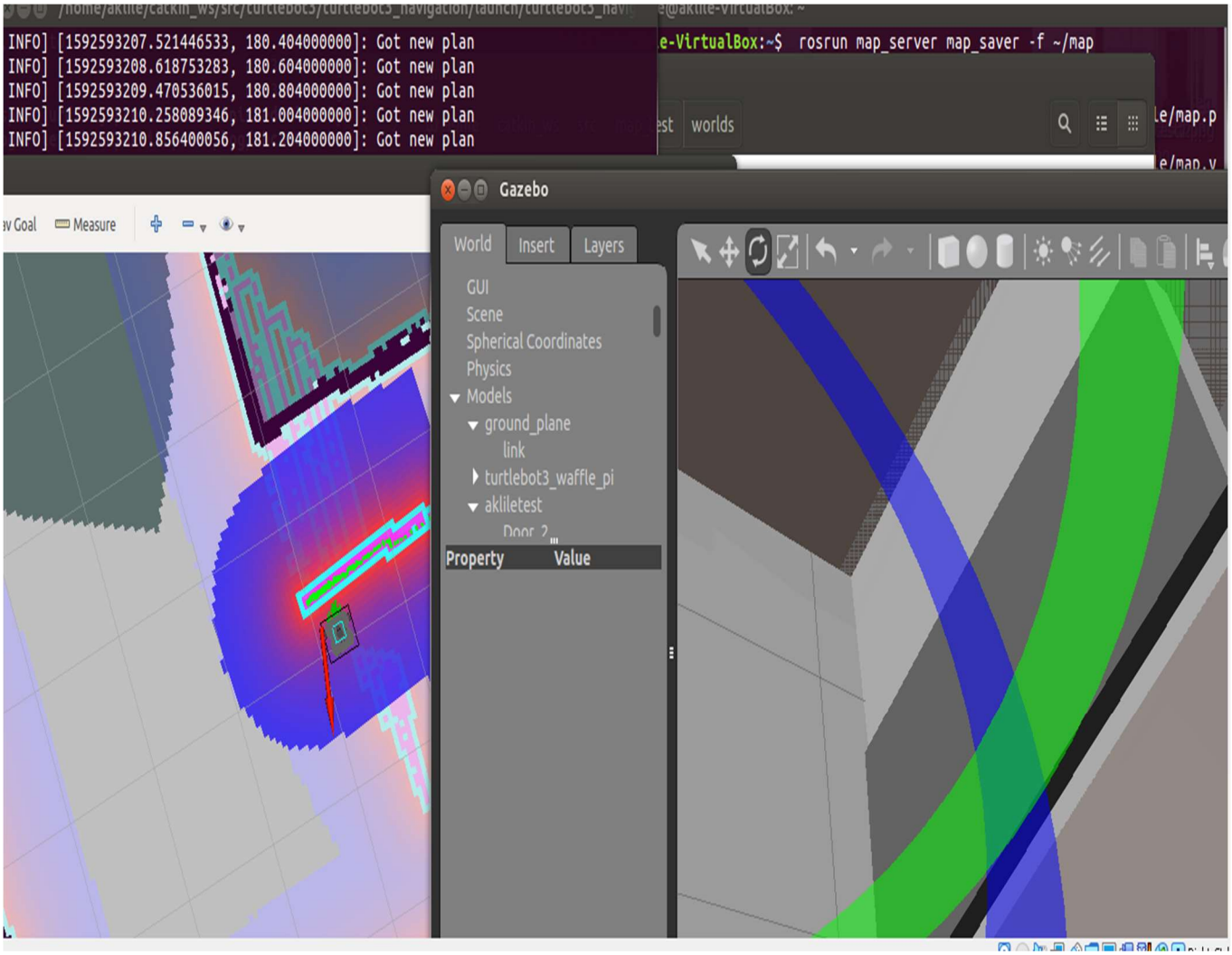


Figure 6.16 Navigation Scenario Visualized by RViz and gazebo model move to the door

6.4.3 Door Detection in simulated world

The following section shows the experiment conducted using camera feed from simulated world and the proposed algorithm.

6.4.3.1 Door detection using robot camera feed in simulated world

As described above to detect the door first getting the camera feed is important before proceeding with detection. To get the camera feed of the simulated world display it I use Image_ View function.

Camera feed of simulated indoor world from different direction is shown below.

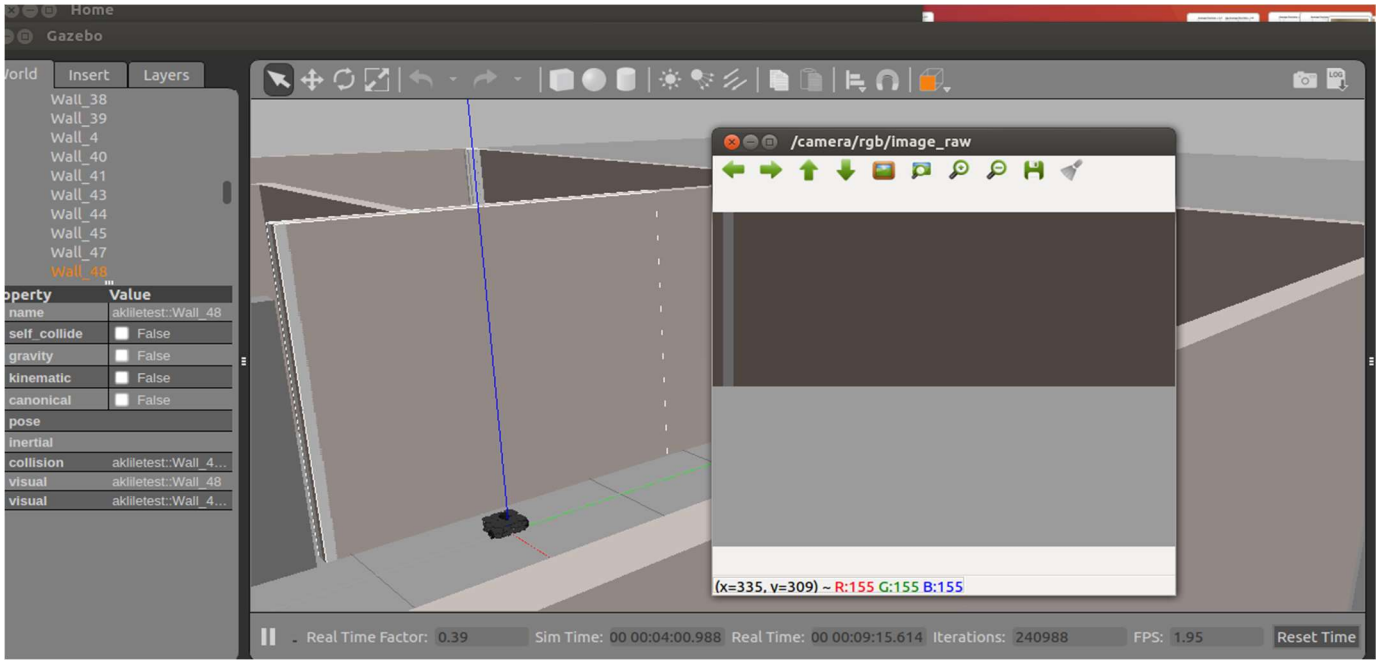


Figure 6.17 Navigation Scenario move to the door Visualized by RViz and Image_ view move to the door

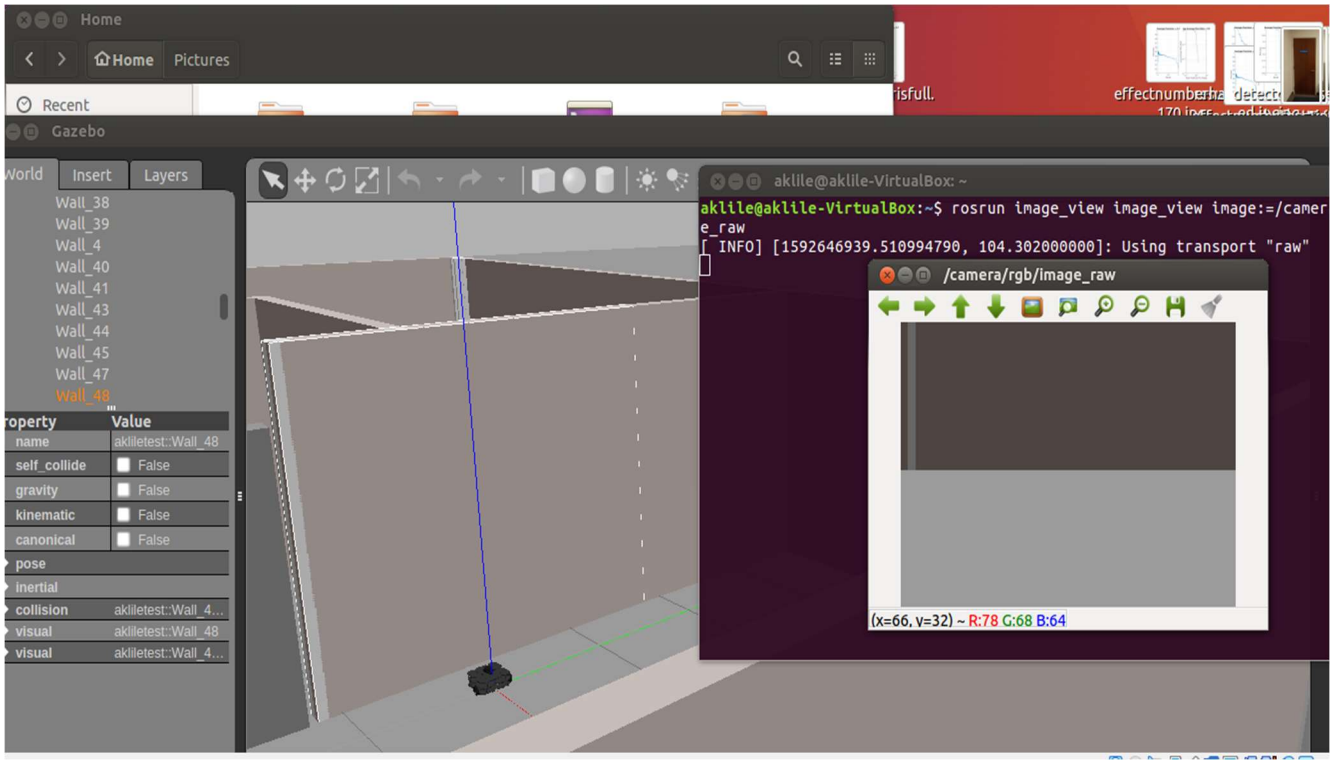
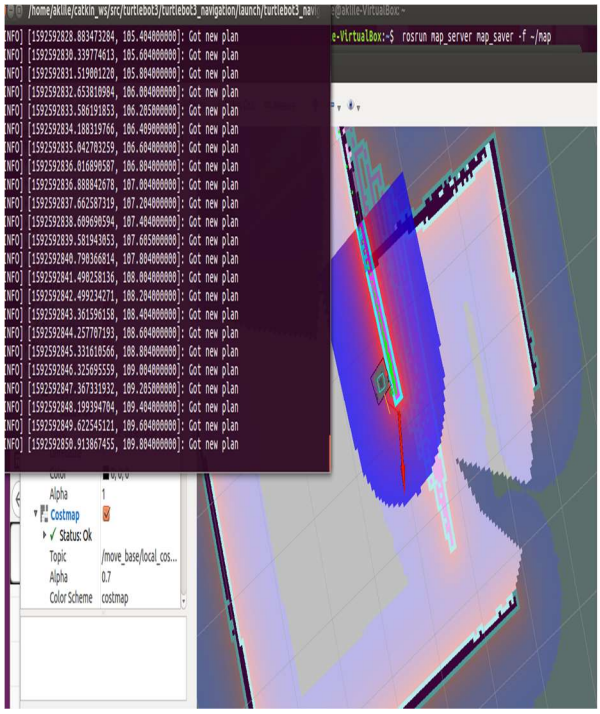


Figure 6.18 *Navigation Scenario move to the door Visualized by RViz and Image_ view move to the door*



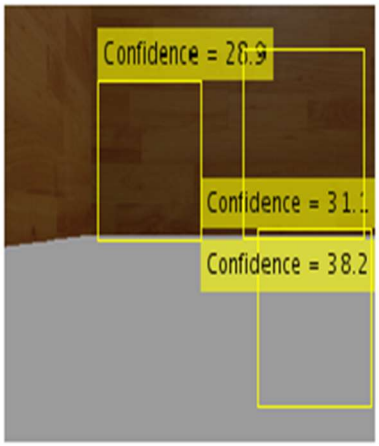


Fig 6.19 Rviz showing door detection and navigation

The above figure shows the result of door detection from robot camera feed together with navigation

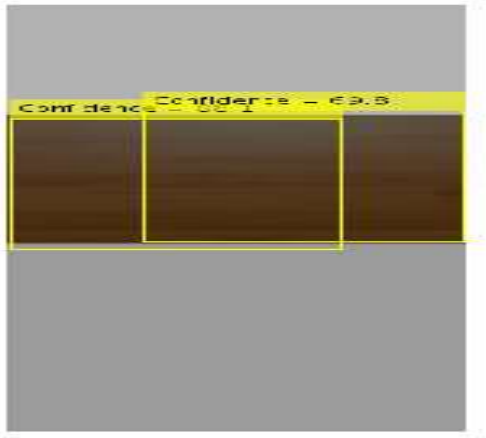


Fig 6.20 additional door detection with robot camera in simulated world

6.4.3.2 cases when Door detection using robot camera feed in simulated world fails

When the texture and the color of the door is very different from what is trained on the algorithm cannot detect doors properly

*Figiure*6.21 when the color and texture of the door is different the algorithm cannot detect it

6.5 Discussion and interpretation of results

As already discussed in these chapter to answer the research question and evaluate the merits of having prior knowledge of spatial layout of the room on the performance of object detection algorithm two major experiments are conducted.

To find out whether having prior knowledge of spatial layout improve object detection accuracy, it is important to conduct two separate experiments. In the first case there would be no spatial knowledge incorporated in the algorithm, where as in the second case spatial knowledge would be incorporated.

In the first experiment Aggregate Channel features (ACF) object detector is trained with images without the room layout recovery algorithm (which recovers surface of the room) and the performance of the detector is evaluated.

In the second experiment first the training images are trained with room layout recovery algorithm. Which recovers at most 5 surfaces and labels each surfaces using different color (floor, left wall, center wall, right wall and ceiling) of the room. Then it extracts the free space between the surfaces and clutter (objects inside room) by drawing a box to show the free space.

Next the output of the room layout recovery algorithm is used as input for Aggregate Channel features (ACF) object detector. The output of the room layout recovery algorithm is then labeled for door, text& sign for training Aggregate Channel features (ACF) object detector. Which means Aggregate Channel features (ACF) object detector has prior knowledge of the spatial layout of the room. Since, the location of each surface is already labeled and known by room layout recovery algorithm.

As shown in these chapter when Aggregate Channel features (ACF) object detector is trained without room layout recovery algorithm it has less accuracy. For example, considering door detection, when Harris detector is trained with 900 images and tested using 300 images, Aggregate Channel features (ACF) object have average precision of 0.8 and Log average miss rate of 0.4(Figure 6.10).

In the second case while considering the effect of having prior knowledge of spatial layout of the room, first the room layout recovery algorithm is trained and the output is used to train Aggregate Channel features (ACF) object detector. For this case Harris detector is trained with 900 images and tested using 300 images as before, which gives an improvement of both values. Average precision of 1.0 and Log average miss rate of 0.0(Figure 6.11) is recorded in the second case, illustrating the benefit of having spatial knowledge to improve object detection accuracy.

The research also shows the benefit of utilizing widely available indoor objects like doors, text and sign to identify the location of the robot. By putting sign and text on doors and hallways the robot can easily identify where it is located. As shown in figure 6.16, figure 6.17, figure 6.18, figure 6.19 and figure 6.20 the robot can create a map navigate towards a goal (door, text &sign) by combining object detection and slam algorithm. The robot can recognize its location by detecting sign and text. Moreover, the robot could easily identify obstacles because It already know the surface it could easily differentiate obstacle from the floor, which improve the navigation ability of the robot.

CHAPTER SEVEN

7. Conclusion and Future Work

7.1 Conclusion

Making mobile robots truly ubiquitous and cohabit with human beings require enhancing robot's ability to understand complex indoor environments. Service robots must perceive and understand complex indoor scenes and be able to recover room layout to better grasp the space and orientation of objects and 3D surfaces in the room. Robots ability to reason about the 3D surface have great implication for navigation and object detection. Robots must also take advantage of widely available information in indoor environment (i.e. text, sign door etc.) to solve long standing navigation problems like Loop closure problem (i.e. robot is unable to recognize a place it has already visited). Since text and sign can be used to uniquely identify a place. This study deals with integration of scene analysis algorithm with object detection algorithm to enhance mobile robot's navigation capability.

The proposed method has two steps. The first step is recovering the spatial layout of a room. Given an image the algorism recovers the geometric surface of the room and clutter together with the free space available in the room. The output classifies and segments the geometric class (i.e. right wall, left wall, center wall, ceiling and floor) of the room labels them with different color. This will be used as an input for Aggregate Channel Features Detector(ACF). The ACF detector is trained using the output of room lay out recovery algorithm. Finally, ACF detector is used to detect land marks (door, sign, and text) in the room and will be used for mobile robot navigation.

To find the optimal size of training images rather than merely increasing the number of images without improving performance of the algorithm ACF is trained with 170,300 and 900 images and the average precision for each one is 0.4, 0.6 and 0.8 respectively. 900 images give a reasonable performance

The effect of using different feature extractors namely Minimum Eigen Value, Harris detector and Surf detector is also measured. Surf performed worse compared with the other two algorithms while Minimum Eigen Value and Harris detector have similar average precision of 0.1 when trained using 130 images.

To test effectiveness of the combination of the two algorithms first ACF object detector is trained with raw images without using output of room layout recovery algorithm. ACF is trained with 900 images using Harris feature extractor and 300 images are used to test the performance. An average precision of 0.8 and log average miss rate of 0.4 is achieved.

Combining room layout recovery and Aggregate channel features, algorithm using Harris feature extractor and 900 images an average precision of 1.0 and log average miss rate of 0.0 is achieved, outperforming the approach that uses only ACF detector.

The proposed algorithm is implemented on turtle bot and in a simulated world. By taking a camera feed from the simulated word the turtle bot was able to detect the doors using proposed algorithm. The robot is able to create a map and navigate the environment and detect objects utilizing the proposed algorithm.

7.2 Future Work

The proposed integration of room layout lay out recovery and aggregate channel features algorithm can be improved by adding other scene analysis algorithms. For instance, by incorporating scene classification algorithm the robot would be able to navigate semantically (i.e. move to kitchen instead of map coordinates). Moreover, integrating RGBD depth cameras also enhance the performance of the algorithm.

To overcome the limitation of the current research due to lack of a fully developed data set in local languages should also be addressed in future research. Incorporating a wide variety of signs would also be beneficial.

Due to time constraints the current research is conducted in simulation. In the future I plan to conduct the experiments using a real robot. Moreover, I would like to explore deep learning techniques for scene analysis and robot navigation.

References

[1] V. R. C. Landsiedel, M. Walter & D. Wollherr, "A review of spatial reasoning and interaction for real-world robotics," *Advanced Robotics,* 19 Jan 2017.

[2] D. D. a. D. K. NITIN KUMAR DHIMAN, "Where am I? Creating spatial awareness in unmanned ground robots using SLAM: A survey," *SADHANA,* vol. Vol. 40, Part 5, pp. pp. 1385–1433., August 2015, .

[3] T. A. a. B. R. V. Bhanu Chander, "Recovering Free Space from a Single Two-Point Perspective Image for Mobile Robot Navigation for Indoor Applications," in *Machines, Mechanism and Robotics, Lecture Notes in Mechanical Engineering, Lecture Notes in Mechanical Engineering,*, 2019, pp. pp 15-26.

[4] J. L. Owens, "Visual Perception For Robotic Spatial Understanding," Publicly Accessible Penn Dissertations2019.

[5] D. H. Varsha Hedau, David Forsyth, "Recovering the Spatial Layout of Cluttered Rooms," *iccv,* 2009.

[6] D. H. Bart Nabbe, Alexei A.A. Efros and Martial Hebert, "Opportunistic Use of Vision to Push Back the Path-Planning Horizon," *iros*.

[7] B. S. Carl Case∗, Adam Coates, Andrew Y. Ng, "Autonomous Sign Reading for Semantic Mapping."

[8] S. F. Shenlong Wang, Raquel Urtasun, "Lost Shopping! Monocular Localization in Large Indoor Spaces."

[9] D. H. A. A. E. M. Hebert, "Geometric Context from a Single Image."

[10] D. H. A. A. E. M. Hebert, "Putting Objects in Perspective," *International Journal of Computer Vision,* vol. 80, pp. 3–15, (2008).

[11] A. K. Congcong Li, Ashutosh Saxena,Tsuhan Chen, "Towards Holistic Scene Understanding:Feedback Enabled Cascaded Classification Models."

[12] Y. L. a. S. T. Birchfield, "Image-Based Segmentation of Indoor Corridor Floors for a Mobile Robot."

[13] A. M. N. Sanchit Aggarwal, C V Jawahar, "Estimating Floor Regions in Cluttered Indoor Scenes from First Person Camera View," *International Conference on Pattern Recognition),* 2014.

[14] Y. L. Zhichao Chen, Stanley T. Birchfield, "Visual detection of lintel-occluded doors by integrating multiple cues using a data-driven Markov chain Monte Carlo process," *Robotics and Autonomous Systems,* vol. 59 pp. 966–976, 2011.

[15] "Towards a Sign-Based Indoor Navigation System for People with Visual Impairments Visual Impairments," *ASSETS. ,* pp. 287–288, 2017.

[16] D. V. D. Prajakta Ganesh Pawar, "Scene Understanding: A Survey to See the World at a Single Glance," *International Conference on Intelligent Communication and Computational Techniques (ICCT),* Sep 28-29, 2019.

[17] S. H. K. z. Muzammal Naseery, Fatih Porikliy, "Indoor Scene Understanding in 2.5/3D for Autonomous Agents: A Survey," *arXiv,* 10 Jan 2019.

[18] S. C. S. Aarthi, "Scene Understanding – A Survey," *IEEE International Conference on Computer, Communication, and Signal Processing (ICCCSP-2017),* 2017.

[19] S. S. a. J. Xiao, "Sliding shapes for 3d object detection in depth images," in *European conference on computer vision*, 2014.

[20] "matworks," ed.

[21] H. B. R. Mottaghi, M. Rastegari, and A. Farhadi, "Newtonian scene understanding: Unfolding the dynamics of objects in static images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[22] D. T. A. Tejani, R. Kouskouridas, and T.-K. Kim, "Latent-class hough forests for 3d object detection and pose estimation," in *European Conference on Computer Vision. Springer,*, 2014.
[23] P. K. B.s. Kim, and S. Savarese, " 3d scene understanding by voxel-crf," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.
[24] B. L. H. Peng, W. Xiong, W. Hu, and R. Ji, " Rgbd salient object detection: a benchmark and algorithms," in *European conference on computer vision*, 2014.
[25] I. E. A. Farhadi, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Computer Vision and Pattern Recognition*, 2009.
[26] H. L. E. Delage, and A. Y. Ng., "A dynamic bayesian network model for autonomous 3d reconstruction from a single," in *CVPR*, 2006.
[27] M. H. D. C. Lee, and T. Kanade., "Geometric reasoning for single image structure recovery," in *CVPR*, 2009.
[28] A. C. Chuhang Zouy, Qi Shanz and Derek Hoiem, "LayoutNet: Reconstructing the 3D Room Layout from a Single RGB Image," in *CVPR*.
[29] A. A. E. Derek Hoiem, Martial Hebert, "Closing the Loop in Scene Interpretation " *CVPR,* 2008.
[30] S. F. D. Lin, and R. Urtasun, "Holistic scene understanding for 3d object detection with rgbd cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.
[31] S. G. A. S. a. D. K. G. Heitz, "Cascaded classification models: Combining models for holistic scene understanding," in *NIPS*, 2008.
[32] A. K. A. S. a. T. C. C. Li, "Towards holistic scene understanding: Feedback enabled cascaded classification models," *Advances in Neural Information Processing Systems,* pp. 1351–1359.,, 2010,.
[33] G. C. BARCELÓ, "Image-Based Floor Segmentation in Visual Inertial Navigation," Stockholm, Sweden2012.
[34] S. L. Fereshteh Sadeghi, "CAD2RL: Real Single-Image Flight without a Single Real Image," *arXiv:1611.04201* Jun 2017.
[35] A. Saxena, "Learning depth from single monocular images," *Advances in neural information processing systems,* p. 2006, 1161–1168.
[36] Nourbakhsh, "Appearance-based obstacle detection with monocular color vision," *AAAI/IAAI,* pp. 866–871, 2000.
[37] d. Croon, "Sky segmentation approach to obstacle avoidance," in *IEEE Aerospace Conference,* , 2011.
[38] A. M. N. Sanchit Aggarwal, C. V. Jawahar, "Estimating Floor Regions in Cluttered Indoor Scenes from First Person Camera View," in *International Conference on Pattern Recognition ICPR 2014*, August 2014.
[39] Y. L. a. S. T. Birchfield, "Image-Based Segmentation of Indoor Corridor Floors for a Mobile Robot."
[40] Y. A. Muhammad Sami, Mohsin Jamil , Syed Omer Gilani, Muhammad Naveed, "Text Detection and Recognition for Semantic Mapping in Indoor Navigation."
[41] B. S. Carl Case, Adam Coates and Andrew Y. Ng, "Autonomous Sign Reading for Semantic Mapping."
[42] Y. T. C. Y. A. A. Xiaodong Yang, "Context-based Indoor Object Detection as an Aid to Blind Persons Accessing Unfamiliar Environments," 210.
[43] X. Y. Yingli Tian, and Aries Arditi, "Computer Vision-Based Door Detection for Accessibility of Unfamiliar Environments to Blind Persons."
[44] D. Anguelov, Koller, D., Parker, E., Thrun, S, "Detecting and modeling doors with mobile robots.," *Proceedings of the IEEE International Conference on Robotics and Automation,* 2004.

[45] D. Kim, Nevatia, R, "A method for recognition and localization of generic objects for indoor navigation.," *ARPA Image Understanding Workshop,* 1994.
[46] F. L. M. S.A. Stoeter, and N. P. Papanikopoulos, "Real-time door detection in cluttered environments," *Int. Symposium on Intelligent Control,* 2000.
[47] D. K. D. Anguelov, E. Parker, and S. Thrun, "Detecting and modelling doors with mobile robots," *IEEE Int. Conf. on Robotics and Automation,* pp. 3777–3784, 2004.
[48] J. M. M. M. J. R. Asensio, and L. Montano, "Goal directeed reactive robot navigation," *In IEEE Int. Conf. on Robotics and Automation,* 2905–2910.
[49] E. A. R. Mu˜noz-Salinas, M. Garcia-Silvente, and A. Gonzales, "Door detection using computer vision and fuzzy logic," *WSEAS Transactions on Systems,* vol. 10, no. 3, pp. 3047–3052, 2004.
[50] W. S. a. J. Samarabandu., "Investigating the performance of corridor and door detection algorithms in different environments.," *Int. Conf. on Information and Automation,* pp. 206–211, 2006.
[51] Z. Chen, Birchfield, S, "Visual Detection of Lintel-Occluded Doors from a Single Image," *IEEE Computer Society Workshop on Visual Localization for Mobile Platforms,* 2008.
[52] R. Munoz-Salinas, Aguirre, E., Garcia-Silvente, M., Gonzalez, "Door-detection using computer vision and fuzzy logic," *Proceedings of the 6th WSEAS International Conference on Mathematical Methods & Computational Techniques in Electrical Engineering,* 2004.
[53] A. C. Murillo, Kosecka, J., Guerrero, J.J., Sagues, C, "Visual door detection integrating appearance and shape cues," *Robotics and Autonomous Systems,* 2008.
[54] X. Y. Yingli Tian, and Aries Arditi, "Computer Vision-Based Door Detection for Accessibility of Unfamiliar Environments to Blind Persons," *ICCHP,* pp. 263–270, 2010.
[55] P. E. R. Juan Fasola, and M Veloso, "Fast goal navigation with obstacle avoidance using a dynamic local visual model," *The VII Brazilian Symposium of Artificial Intelligence,* 2005.
[56] R. A. B. Liana M Lorigo, and WEL Grimsou, "Visually-guided obstacle avoidance in unstructured environments," *Intelligent Robots and Systems,* 1977.
[57] I. U. a. I. Nourbakhsh, "Appearance-based obstacle detection with monocular color vision," *AAAI/IAAI,* pp. 866–871, 2000.
[58] M. R. B. S. Riisgaard, "SLAM for Dummies : A Tutorial Approach to Simultaneous Localization and Mapping.," 2000.
[59] A. Aga, " Avoidable visual impairment among elderly people in a Slum of Addis Ababa," *The Ethiopian Journal of Health Development*.
[60] T. B. Hugh Durrant-Whyte, "Simultaneous Localisation and Mapping (SLAM):Part I The Essential Algorithms," *IEEE Robotics & Automation Magazine,* vol. 13 no. 2 2006.
[61] N. T. a. G. B. Ardhisha Pancham, "Literature Review of SLAM and DATMO," in *Robotics and Mechatronics Conference of South Afric*, South Africa, 2011.
[62] S. R. a. M. R. Blas, *SLAM for Dummies :A Tutorial Approach to Simultaneous Localization and Mapping* 2005.
[63] M. I. Ribeiro, *Kalman and Extended Kalman Filters:Concept, Derivation and Properties*. Institute for Systems and Robotics, 2004.
[64] X. J. T. T.J. Chong, C.H. Leng, M. Yogeswaran, "Sensor Technologies and Simultaneous Localization and Mapping," *IEEE International Symposium on Robotics and Intelligent Sensors (IRIS 2015),* vol. 76, pp. 174 – 179, 2015
[65] C. C. a. L. C. a. H. C. a. Y. L. a. D. S. a. J. N. a. I. R. a. J. J. Leonard, "Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age}," *IEEE Transactions on Robotics,* vol. 32, no. 6, pp. 1309–1332, 2016.
[66] K. B. Tsai Grace, "Michigan Indoor Corridor Dataset."

[67] A. B. J. a. G. Sohn, "GEOMETRIC CONTEXT AND ORIENTATION MAP COMBINATION FOR INDOOR CORRIDOR MODELING USING A SINGLE IMAGE," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. Volume XLI-B4, 2016.

[68] M. H. a. T. K. David C. Lee, "Geometric Reasoning for Single Image Structure Recovery," *IEEE Conference on Computer Vision and Pattern Recognition,* 2009.

[69] A. A. E. A. M. H. DEREK HOIEM, "Recovering Surface Layout from an Image," *International Journal of Computer Vision* vol. 75(1), pp. 151–172, 2007.

[70] J. G. E. B. L. Del Pero, J. Schlecht, K. Barnard, "Sampling Bedrooms," *CVPR,* 2011.

[71] D. F. L. Del Pero J. Bowdish, B. Kermgard, E. Hartley, K. Barnard,, "Bayesian geometric modeling of indoor scenes," *CVPR,* 2012.

[72] H. I. El-Zorkany, "Robot Programming," vol. 23, no. 4, 1984.

[73] S. S. D. Hoiem, *REPRESENTATIONS AND TECHNIQUES FOR 3D OBJECT RECOGNITION AND SCENE INTERPRETATION*.

[74] R. A. Piotr Doll ´ ar, Serge Belongie, and Pietro Perona, "Fast Feature Pyramids for Object Detection," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*.

Appendix I

Sample code SPATIALLAYOUT

```
function [ boxlayout,surface_labels ] = getspatiallayout(imdir,imagename,workspcdir)
% GETSPATIALLAYOUT Given an image, estimate its spatial layout, consisting
%   of a box layout and pixel labels of different surfaces. Please refer to
%   the readme file provided with this software for detailed meaning of
%   these outputs.
%
% USAGE: [ boxlayout,surface_labels ] =
%              getspatiallayout(imdir,imagename,workspcdir)
%
% INPUT:
%   imdir - directory containing the original image to be processed. Use '/'
%      (not '\') to separate directories.
%   imagename - original image name
%   workspcdir - directory to hold intermediate results. Internally, two
%      independent directories are created inside this directory to hold
%      visualizations and data files.
%
% OUTPUT:
%   binlayout - structure containing the estimated approximation of the
%      indoor scene as a 3D box, with the following fields -
%   .polyg - cell array of size (n x 5), each containing coordinates of
%      planar surfaces of the room (left, right, middle walls, and, floor
%      and ceiling). The (i,j)-th cell entry corresponds to i-th box
%      layout hypothesis, and j-th plane of that hypothesis.
%   .init - An array of size (n x 2). The i-th row (score, index) represents
%      score and index of a hypothesis in the polyg cell array
```

```
%   surface_labels - structure containing the per pixel likelihood of each of
%       the surfaces, and objects, with following fields -
%   .init - An array of size (m x 7), where m is th total number of
%       superpixels detected in the image. The (i,j)-th entry represents
%       the likelihood of the i-th superpixel being labeled as j-th
%       surface.
outimgdir = [workspcdir 'Images/'];
if ~exist(outimgdir,'dir')
   mkdir(outimgdir);
end
workspcdir = [workspcdir 'data/'];
if ~exist(workspcdir,'dir')
   mkdir(workspcdir);
end

% tempdir='./sptiallayouttempworkspace/';
%Compute vps
boxlayout=[];
surface_labels=[];

[vp p All_lines]=getVP(imdir,imagename,0,workspcdir);
img=imread([imdir imagename]);
[h w kk]=size(img);
VP=vp;
if numel(VP)<6
   return;
end
```

```
vp=[VP(1) VP(2);VP(3) VP(4);VP(5) VP(6)];
[vp P]=ordervp(vp,h,w,p);
[vv linemem]=max(P,[],2);
vpdata.vp=vp;
vpdata.lines=All_lines;
vpdata.linemem=linemem;
vpdata.dim=[h w];
%visvp(vpdata,img)
%Get segmentation and GC surface confidence maps
% sigma=num2str(0.8);
% k1=num2str(100);
% min1=num2str(100);
% inputim=[workspcdir imagename(1:end-4) '.ppm'];
% outputim=[workspcdir imagename(1:end-4) '.pnm'];
% [s w]=system(['./segment' ' ' sigma ' '  k1  ' ' min1 ' ' inputim ' ' outputim ]);

%visvp(vpdata,img)
%Get segmentation and GC surface confidence maps
%  sigma=num2str(0.8);
%  k1=num2str(100);
%  min1=num2str(100);
%  inputim=[workspcdir imagename(1:end-4) '.ppm'];
%  outputim=[workspcdir imagename(1:end-4) '.pnm'];
% [s w]=system(['./segment' ' ' sigma ' '  k1  ' ' min1 ' ' inputim ' ' outputim ]);
```

```
segext='pnm';
nsegments=[5 15 25 35 40 60 80 100];
%fn=['../Imsegs/' imagename(1:end-4) '.' segext];
fn=['/home/aklile/Documents/MATLAB/varsha_spatialLayout/SpatialLayout/Imsegs/'
imagename(1:end-4) '.' segext];
imseg = processSuperpixelImage(fn);

tic
imdata = mcmcComputeImageData(im2double(img), imseg);% made changes here
toc

load(fullfile('../LabelClassifier/', 'Classifiers_gc.mat'));

spfeatures = mcmcGetAllSuperpixelData(imdir, imseg);
[efeatures, adjlist] = mcmcGetAllEdgeData(spfeatures, imseg(1));

nsegments=[5 15 25 35 40 60 80 100];

pE[10] = test_boosted_dt_mc(eclassifier, efeatures[10]);
pE[10] = 1 ./ (1+exp(ecal(1)*pE{1}+ecal(2)));
smaps{1} = msCreateMultipleSegmentations(pE{1}, adjlist{1}, ...
   imseg(1).nseg, nsegments);

for k = 1:numel(nsegments)
   if max(smaps{1}(:, k))>0
      segfeatures{1, k} = mcmcGetSegmentFeatures(imseg, ...
         spfeatures{1}, imdata, smaps{1}(:, k), (1:max(smaps{1}(:, k))));
   end
end
```

```
%Get surface label confidences initial from GC
normalize = 1;
pg=zeros(imseg.nseg,7);%7 labels
%get P(L/I)
pg = msTest(imseg, segfeatures, smaps, ...
   labelclassifier, segclassifier,normalize);

filename=fullfile(workspcdir, [imagename(1:end-4) '_lc_gc.mat' ]);
save(filename,'pg');

%visualize
%Compute intergral images for features
tic
[integData]=getIntegralimages([imagename],vpdata,imseg,500,workspcdir,imdir);
toc

%Get candidate layouts and their features
[polyg, Features] = getcandboxlayout( vpdata.vp,vpdata.dim(1),vpdata.dim(2),integData);

%Get initial estimate
Features1=Features;
load ../LearntClassifiers/pf_i.mat
lay_score=[];
numL=size(Features,1);
% change features
if(numel(weights)==14)
   tmpf1=sum(Features(:,1:5).*Features(:,11:15),2);
   tmpf2=sum(Features(:,1:5).*Features(:,16:20),2);
```

```
    tmpf3=sum(Features(:,6:10).*Features(:,11:15),2);
    tmpf4=sum(Features(:,6:10).*Features(:,16:20),2);
    Features=[Features(:,1:10) tmpf1 tmpf2 tmpf3 tmpf4];
end

%evaluate
score=repmat(weights,[numL,1]).*Features;
score=sum(score,2);
[vv ii]=sort(score,'descend');

boxlayout.polyg=polyg;
boxlayout.init=[vv ii];

load ../LearntClassifiers/pf_il.mat
Features=Features1;
if(numel(weights)==59)
    tmpf1=sum(Features(:,1:5).*Features(:,11:15),2);
    tmpf2=sum(Features(:,1:5).*Features(:,16:20),2);
    tmpf3=sum(Features(:,6:10).*Features(:,11:15),2);
    tmpf4=sum(Features(:,6:10).*Features(:,16:20),2);
    Features=[Features(:,1:10) tmpf1 tmpf2 tmpf3 tmpf4 Features(:,31:75)
];%Features(:,114:122)];

end
score=repmat(weights,[numL,1]).*Features;
score=sum(score,2);
[vv ii]=sort(score,'descend');
```

```
boxlayout.reestimated=[vv ii];
lay_scores=[vv ii];
save([workspcdir imagename(1:end-4) '_layres.mat'],'polyg','lay_scores');%,'avg_pg');

figure(101);
drawnow;
for lay=1:25
   layoutid=ii(lay);
   Polyg=[];
   for fie=1:5
      Polyg{fie}=[];
      if size(polyg{layoutid,fie})>0
         Polyg{fie}=polyg{layoutid,fie};
      end
   end

   tempimg=displayout(Polyg,w,h,img);
   subplot(5,5,lay);imshow(uint8(tempimg),[]);title(num2str(vv(lay)));
end
saveas(101,[outimgdir imagename(1:end-4) '_boxlayouts.png']);

Polyg=[];
for fie=1:5
   Polyg{fie}=[];
   if size(polyg{ii(1),fie})>0
      Polyg{fie}=polyg{ii(1),fie};
   end
end
```

```
ShowGTPolyg(img,Polyg,103);
saveas(103,[outimgdir imagename(1:end-4) '_boxlayout.png']);


%Re-Compute Surface labels (GC+box layout features)
load(['../LabelClassifier/' 'Classifiers_stage2.mat']);
tic
xspfields=[];%save per image
numSup=imseg.nseg;
for lay=1:numL %each layout
   xspf=[];
   for  supno=1:numSup
      tempbndy=imdata.tracedbndy{supno}{1};
      if size(tempbndy,1) > 30

         YY1=tempbndy(1:10:end,1);XX1=tempbndy(1:10:end,2);
      else

         YY1=tempbndy(:,1);XX1=tempbndy(:,2);
      end

      for fi=1:5  %each field
         xarea=0;
         if size(polyg{lay,fi},1)>0

            XX2=polyg{lay,fi}(:,1);
            YY2=polyg{lay,fi}(:,2);
```

```
            [in on]=inpolygon(XX1,YY1,[XX2;XX2(1)],[YY2;YY2(1)]);

            if numel(find(in==1))==length(in)
               X0=XX1;Y0=YY1;
               xarea=polyarea([X0;X0(1)],[Y0;Y0(1)]);
            elseif  numel(find(in==1))==0

               X0=[];Y0=[];
               xarea=0;
            else
               xarea=polyintarea(XX1,YY1,XX2,YY2,0);

            end


         end
         xspf(supno,fi)=xarea;
      end

   end
   xspfields{lay}=xspf;
end
toc
```

```
nsp=imseg.nseg; %num of suppixels
smap = [1:nsp];
smaps{1} = smap(:);

clear segfeatures
for k = 1: 1%numel(nsegments)
   features = mcmcGetSegmentFeatures(imseg, ...
      spfeatures{1}, imdata, smaps{1}(:, k), (1:max(smaps{1}(:, k))));

   for lay=1:numL
      tempfeatures=features;
      [fieldfeatures]=segfieldsfeat_sup(imseg,imseg.nseg,smaps{1}(:,k),xspfields{lay});
      tempfeatures(:,95:100)=fieldfeatures(1:size(features,1),:);
      tempfeatures(:,101:106)=pg{1}(:,1:6);
      segfeatures{lay,k}  = tempfeatures;
   end
end

if numel(vv)>100
   A = [vv(1) 1;vv(100) 1];
else
   A = [vv(1) 1;vv(end) 1];
end
b = [0;log(49)];
params = inv(A)*b;
```

```
%     conf = 1./(1+exp(params(1)*vv(1:5)+params(2)));
conf = 1./(1+exp(params(1)*vv(1)+params(2)));
conf=conf./sum(conf);

avg_pg=zeros(imseg.nseg,7);%7 labels

tic
for j=1:1 %take best 5 scored layouts
   lay=ii(j);

   %get P(L/F,I)
   pg = msTest(imseg, segfeatures(lay, :), smaps, ...
      labelclassifier);%, segclassifier,normalize);
   avg_pg=avg_pg+pg{1}.*conf(j);

end
toc

filename=fullfile(workspcdir,[imagename(1:end-4) '_lc_st2.mat' ]);
save(filename,'avg_pg');

surface_labels.restimated={avg_pg};
surface_labels.init=pg;
rl=[255  255 0  0 0 255 0];
gl=[255 0 255  255 0 0 0];
bl=[0  0 255 0 255 255 0];
rl1=[255 202 132 255 0 255 0 191 0 0];
gl1=[236 255 112 255 191 0 255 21 0 0];
```

```
bl1=[139 112 255  255 255 0 0 133 255 0];
cimages = msPg2confidenceImages(imseg,{avg_pg});
[aa, indd]=max(cimages{1}(:,:,1:6),[],3);

figure(102);
clear mask_color;
mask_r = rl(indd);
mask_g = gl(indd);
mask_b = bl(indd);
mask_color(:,:,1) = mask_r;
mask_color(:,:,2) = mask_g;
mask_color(:,:,3) = mask_b;

hsvmask=rgb2hsv(mask_color);
hsvmask(:,:,3)=aa*255;
%     hsvmask(:,:,2)=aa;
mask_color=hsv2rgb(hsvmask);

tempimg = double(img)*0.5 + mask_color*0.5;
%         tempimg =  mask_color;
imshow(uint8(tempimg));
saveas(102,[outimgdir imagename(1:end-4) '_surfacelabels.png']);

end
```