Scene Interpretation in Images and Videos.

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science (by Research Dual Degree)
in
Computer Science

by

Chetan J
200402009
chetan@research.iiit.ac.in

Center for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2010

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Scene Interpretation in Images and Videos” by Mr. Chetan. J, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Dr. C. V. Jawahar,	Dr. Madhava Krishna

To CVIT, IIIT Hyderabad.

To my Family.

Acknowledgments

I would like to thank Dr. C.V. Jawahar and Dr. Madhava Krishna for their dedication, support and guidance throughout my research over the last few years.

I would also like to thank my fellow lab mates at the Center for Visual Information Technology (CVIT), IIIT Hyderabad, who had made great contribution by sharing ideas, criticisms and materials. I specially thank Visesh, Paresh, Ranjeeth, Anil, Sreekanth, Rakesh, Rasagna, Pradeep, Neeba, Pradhee, Chandrika, Karthika, ,Supreeth, Prachi, Pratyush, Chhaya, Mihir, Omkar, Vidyadhari, Priyanka, Gopal, Jinesh, Vibhav, Pavan. D and Abhijit for their valuable suggestions, kindness and strong moral support. The financial support I received from the CVIT during my Masters studies is acknowledged.

Above all I am thankful to my parents, my relatives, faculty in IIIT-H and all those from CVIT and other research labs who had at some point in time helped me with their unconditional support and unbounded patience.

June 26, 2010

Abstract

Scene interpretation is a fundamental task in both computer vision and robotic systems. We deal with two important aspects of scene interpretation, they are scene reconstruction and scene recognition. Scene reconstruction is determining 3D positions of world points and retrieving camera poses from images. It has several applications such as virtual building editing in computer aided architecture, video augmentation in film industry and planning and navigation in mobile robotics. Among several approaches to modeling the scene, we deal with piecewise planar modeling due to several advantages: Man-made environments are often piece-wise-planar, planar modeling has compact representation and this can be easily modified. We propose a convex optimization based, approach for piecewise planar reconstruction. We show that the task of reconstructing a piece-wise planar environment can be set in an L_∞ based Homographic framework that iteratively computes scene plane and camera pose parameters. Instead of image points, the algorithm optimizes over inter-image homographies. The resultant objective function is minimized using Second Order Cone Programming algorithms. Apart from showing the convergence of the algorithm, we also empirically verify its robustness to error in initialization through various experiments on synthetic and real data. We intend this algorithm to be in between initialization approaches like decomposition methods and iterative non-linear minimization methods like Bundle Adjustment.

Scene recognition in robotics, specifically terrain scene recognition is one of the fundamental tasks of autonomous navigation. Navigable terrains are examples of planar scenes. The goal of terrain recognition is to recognize various terrains that occur in urban and rural environments in an automated fashion. It has applications in various domains such as advanced driver assistance systems, remote sensing, etc. Various sensing modalities such as ladars, lasers, accelerometers, stereo cameras, omni-directional cameras or combination of them are used in literature. This thesis attacks the problem of scene interpretation using a single camera. This investigation is especially crucial since cameras are relatively low in cost, consume low power, light weight and have the potential to provide very rich information about the environment. Recent advances in computer vision, machine learning and improvements in hardware capabilities have greatly increased the scope of monocular camera, even in unstructured and real world environments. In this thesis, we start with empirical study of promising color, texture and their combination with classifiers such as Support Vector Machines (SVM) and Random Forests. We present comparison across features and classifiers. Then we present a monocular camera based terrain recognition scheme called Partition based classifier. The uniqueness of the proposed scheme is that it inherently incorporates spatial smoothness while segmenting an image, without the requirement of any additional post-processing.

The algorithm is fast because it is build on top of a Random Forest classifier. The efficacy of the proposed solution can be seen as we reach low error rates on both our dataset and other publicly available datasets.

Further partition classifier is extended to be online and adaptive. The new scheme consists of two underlying classifiers. One of which is learnt over bootstrapped or offline dataset, the second is another classifier that adapts to changes on the fly. Posterior probabilities of both the static and online classifiers are fused to assign the eventual label for the online image data. The online classifier learns at frequent intervals of time through a sparse and stable set of tracked patches, which makes it lightweight and real-time friendly. The learning which is acuted at frequent intervals during the sojourn significantly improves the performance of the classifier vis-a-vis a scheme that only uses the classifier learnt offline. The method finds immediate applications for outdoor autonomous driving where the classifier needs to be updated frequently based on what shows up recently on the terrain and without largely deviating from those learnt offline.

Contents

1 Introduction **2**

1.1 Scene interpretation in Computer Vision 2

1.2 Scene interpretation in Robotics 5

1.3 Problem statement and Contributions 7

1.4 Organization of thesis 8

2 Background **10**

2.1 Geometry of Planar Scenes 10

2.1.1 Homography 10

2.1.2 Homographies and Camera parameters 12

2.1.3 Homography Decomposition 12

2.1.4 SFM and 3D reconstruction 14

2.1.5 Bundle Adjustment 15

2.2 Layer extraction 15

2.3 Convex Optimization 19

2.4 Terrain Classification 20

2.4.1 Literature review 20

2.4.2 Features 24

2.4.3 Classifiers 25

3 Piece-wise planar scene reconstruction **30**

3.1 Introduction 30

3.1.1 Contributions 31

3.1.2 Organization 31

3.2 Technical Background 31

3.3 Homographic Framework for Planar Reconstruction 33

3.3.1 SVD based Techniques . . . 33
3.3.2 Implementation Issues and Sensitivity Analysis . . . 34
3.4 Convex Framework for Planar Reconstruction . . . 36
3.4.1 Formulation of the Objective Function . . . 36
3.4.2 Proposed Algorithm . . . 37
3.4.3 Discussions . . . 38
3.4.4 Additional Constraints . . . 40
3.4.5 Issues with Rotation and Normal . . . 41
3.5 Experimental Analysis . . . 42
3.5.1 Synthetic Data . . . 42
3.5.2 Real Data . . . 46
3.6 Discussion . . . 48

4 Terrain recognition using monocular camera 53
4.1 Introduction . . . 53
4.1.1 Contributions . . . 53
4.2 Problem Parameters . . . 54
4.2.1 Features . . . 54
4.2.2 Classifiers . . . 54
4.3 Data set and Experimental Setting . . . 55
4.3.1 Data set . . . 55
4.4 Classification procedure . . . 57
4.5 Experiments . . . 58
4.5.1 Experiment 1: Comparison accros classifiers . . . 58
4.5.2 Experiment 2: Effect on number of dimensions . . . 60
4.5.3 Experiment 3: Effect on patch size . . . 60
4.6 Discussion . . . 61

5 Fast and Adaptive Terrain recognition 62
5.1 Introduction . . . 62
5.1.1 Contributions . . . 62
5.2 Partition based algorithm . . . 63
5.3 Experiments . . . 63
5.3.1 Experiment 1: Comparison with baseline classifiers . . . 64
5.3.2 Experiment 2: Effect on number of Classifier-sets (N) . . . 64
5.3.3 Experiment 3: Spatial smoothness test . . . 67

5.3.4 Discussion 67
5.4 From Image to Video 69
5.5 Adaptive method 69
5.5.1 Performance Gain Due to Adaptive Method 72
5.6 Discussion 75

6 Conclusions and Future Work **77**
6.1 Future work 78

Related Publications **80**

List of Figures

1.1 On the left side, the input images are shown and on the right side we have the desired reconstructed model. 3

1.2 Few examples of Man-made planar scenes, planes are marked with green borders. 4

2.1 Homography induced between two images x and x'. Image courtesy [1] 11

2.2 Left: Consecutive frames from a garden sequence, Right: Sub-images or layers in the video. Image courtesy [2] 16

2.3 A quasiconvex function *Q* on $\mathcal{R}$ which is not convex. It is plotted with a convex function em C. However, any horizontal line slices *Q* in atmost two points, thus creating only convex sublevel sets. 18

2.4 First row: Sample frames from our dataset. Second row: Desired output 21

2.5 Sample raw acceleration data for various different types. Observe that except for grass other terrain record very similar measurements. Image courtesy Weiss *et. al.* [3] 22

2.6 Demonstration of near-far learning using stereo-camera or lasers to obtain dense data. Image courtesy Michael *et al. [4]* 24

2.7 (a) LM filters has a total of 48 filters, which are 2 Gaussian derivative filters at 6 orientations and 3 scales, 8 Laplacian of Gaussian filters and 4 Gaussian filters. (b) Basic Linear Binary Pattern operator 26

2.8 Random Forests, containing two types of nodes in every tree, at each un-filled node, a decision function $f(x)$ is defined on random subspace $x \in X$, where X is feature vector, and the filled node is the class label. Majority voted label from all the trees is the final label of the random forest. 29

3.1 (a,b,c) Plot the $L2$ and L_∞ errors in the rotation angles, translation direction and normal direction respectively. Also are plotted the maximum error ranges for these quantities. The translation and normal direction errors are computed as Euclidean distances in polar space. 35
3.2 Proposed algorithm: Each dot in the above figure represents a homography H_i^j . An iteration for refining i the pose of a single view minimizes over data from all the planes, and an iteration for refining a single planes parameters minimizes over data from all the views. 37
3.3 Plot of L_2 and L_∞ norms of the distance in pose space between estimated and ground truth quantities from Algorithm 1 against increase in variance of Gaussian error in point correspondences. Comparison with two SVD based methods is shown. 43
3.4 Plot of minimum, average and maximum of L_2 norms of the distance in pose space between estimated and ground truth quantities from Algorithm 1 from 100 trails against increase in variance of Gaussian error in point correspondences. 43
3.5 Plot of L_2 norm of the distance in pose space between estimated and ground truth quantities from Algorithm 1 and Bundle adjustment against increase in variance of Gaussian error in point correspondences. 44
3.6 The above figures plot the effect of planes on the accuracy in estimation of the translation and normal parameters respectively. In this experiment we varied the number of planes from 2 to 10 and the number of views was kept constant at 10 . . 45
3.7 The above figures plot the effect of views on the accuracy in estimation of the translation and normal parameters respectively. In this experiment we varied the number of views from 3 to 15 and the number of planes was set to be 3. 45
3.8 (a) Shows the improvement of estimating translation parameters using additional constraints, when a single plane has bad homographies. (b) Shows the estimation accuracy of rotation parameters, using the branch and bound algorithm. The estimation is accurate and robust. 46
3.9 The addition of inter-image homography based constraints improves the robustness of the system. The current cost function is designed to overfit outliers. In the above figure, while the red circle represents the minima corresponding to the error function, the actual global minima, the green triangle represents the global minima while the brown star represents the solution with constraints. Each of the circles represents constraints, and the accuracy of the resultant solution depends on their tightness. . . 47

3.10 ***Dataset 1:*** Oxford-house dataset (a) Sample image from the dataset. (b-c) Plots of the L_∞ error between plane and pose parameters with respect to the ground truth L_2 error shows similar plots. 48
3.11 ***Dataset 2:*** Oxford-corridor dataset (a) Sample image from the dataset . (b-c) Plots of the L_∞ error between plane and pose parameters with respect to the ground truth L_2 error shows similar plots. 49
3.12 ***Dataset 3:*** Synthetic house (a-b) Sample images from the dataset . (c-d) Illustrates the accuracy of our reconstruction. The ground truth and reconstructed models are overlapping to a greater extent 50
3.13 ***Dataset 4:*** UNC dataset (a-b) Sample images from the dataset . (c-d) Texture mapped reconstructions of UNC dataset. 51

4.1 Monocular camera attached at the top of the Van. 55
4.2 Overview of the dataset. 56
4.3 Patches from each of the identified classes. 57
4.4 (a) Test image (b) Ground truth image (c) Labelled image using baseline RF classifier 59
4.5 Experiment with increasing dimensions of combined feature on various classifiers. 60
4.6 Error variation with varying window size. 61

5.1 Pictorial representation of partitioning the images into 4,9 and 16 partitions respectively. 64
5.2 (a) Comparison of base-line classifiers with Partition-based algorithm operated over them. (b)Error rates by using multiple classifier sets. 66
5.3 (a) Test image (b) Characterization by RF classifier (c) Characterization by Partition based method 67
5.4 Test images blended with predicted classifications 68
5.5 Tracked patch-labels across three frames. 70
5.6 Block diagram of the proposed scheme 70
5.7 Test images marked with red-colored-patches, representing the labels that are correctly labelled by Adaptive algorithm but wrongly labelled by offline Partition method. 74
5.8 1_{st} row: Path navigated by robot in a closed loop, marked in green color. 2_{nd} row: Test image with predicted labelled images from the first and second loops. 75

List of Tables

4.1 Base line error-rates on Our dataset and two datasets of Procopio et al. [4]. 58

5.1 1^{st} and 2^{nd} column represents percentage errors of RandomForest(RF) and our partition based algorithm(PM). 3^{rd} and 4^{th} column represents smoothness-error, which corresponds to experiment-3. 5^{th} and 6^{th} column represents the percentage of images, that were labelled just by using Temporal-label-transfer method in Section 5.4, where AVG: Average of percentages of portion of labels that are transferred over sequence of 100 images and Err: Error in label transfer 66

5.2 Comparison of Adaptive algorithm with Offline-partition-based-method 73

Chapter 1

Introduction

The ultimate aim of a robotic vision system is to navigate in a world of realistic complexity. This involves interpreting the scene, objects and events to perform appropriate actions. Scene interpretation is a fundamental task in both computer vision and robotic systems. Though humans and animals are good at scene interpretation, accurate scene interpretation is surprisingly difficult. The difficulty comes mainly due to huge viewpoint changes, clutter, variation in illumination caused by shadows, etc. There are different aspects for scene interpretation in literature. One aspect deals with inferencing depth of the scene which answers the questions like *"What region of the image is near and what region is far ?"* OR *"What region of image is at ground level ?"*. Yet another branch deals with the problem of detecting particular object in the scene containing several objects, which answers the questions like *"Does the scene contain car?"* and *"Where is the car in the image ?"*. Other aspects of interpretation are related to recognition which answers the questions like *"Is it the image of a car?"*, *"What car is it?"*, *"Is it a Tata Sumo ?"*, etc.

In this thesis, we concentrate on two important aspects of scene interpretation. They are scene reconstruction and scene recognition. Scene can be interpreted by a variety of methods and this depends on the kinds of sensors used. Using monocular vision sensors for scene interpretation is the main aim of this research.

1.1 Scene interpretation in Computer Vision

In this section, we discuss the scene reconstruction aspect of scene interpretation in computer vision. The 3D scene reconstruction of rigid scenes from photographic images is one of the most challenging problems in Computer Vision and Photogrammetry. This is a classical problem with both theoretical and practical interest, for example virtual building editing in computer aided ar-

Figure 1.1: On the left side, the input images are shown and on the right side we have the desired reconstructed model.

chitecture and video augmentation in the film industry. Figure 1.1 shows the images of the house model with markings on planes and the desired reconstruction. In robotics, the reconstruction or structure helps the robot to understand what objects or part of the scene is near and what objects are far. For a navigating robot it is an essential task. For example, consider a robot navigating in an environment, where the decision of navigating forward depends on whether the robot has enough room in front of it. 3D reconstruction is essentially retrieving camera poses and determining the 3D positions of world points given their images.

There exist a wide variety of approaches to the image-based modeling problem, for example see [5–19]. The main difference among these methods is the representation of the scene they employ. For instance, Kutulakos and Seitz use voxels [12], Strecha *et al.* use a depth map [18], Gargallo and Sturm use multiple depth maps [9], Baillard and Zisserman use a set of planes [5], while Debevec *et al.* use a combination of those [6]. The most appropriate representation depends on the type of scene that is to be reconstructed and the application that is in consideration.

The planar model is motivated by the following reasons. First, man-made environments are often composed of piecewise planar (See Figure 1.2 containing planar objects such as buildings, cars, indoors, machinery etc.,) or nearly-planar primitives [5–7, 20] and are thus modeled as such to a reasonable degree of approximation. Second, this is a very constrained, compact representation that is thus very stable, and allows one to make the reconstruction process automatic. Third, this representation allows one to modify the reconstruction very easily, i.e. by adding, removing or augmenting objects.

Figure 1.2: Few examples of Man-made planar scenes, planes are marked with green borders.

Most of the existing systems are semi-automatic, based on a three-stage process, e.g. [6, 13, 21]. First, a sparse 3D reconstruction of features (points, lines, etc.) as well as cameras is performed automatically using Structure-from-Motion techniques [22, 23]. Secondly, scene model is chosen and final stage is to estimate its parameters. The first stage is achieved by clustering reconstructed features into higher level geometric primitives such as cubes by e.g. marking edges in the input images. The second stage consists of optimizing the quality of the model parameters by e.g. minimizing the disparity between marked and predicted edges. This approach has proven to give highly photo realistic results, but becomes computationally costly as the scene considered grows in complexity.

Scene surface is modeled as a set of triangles in [15, 16]. The most likely triangulation with respect to the input images is computed using edge swaps from an initial solution obtained using a Delaunay triangulation. However the process is not guaranteed to converge to the global optimum. Here, piecewise planarity is not considered, which creates a non photo realistic reconstructions. Representing a scene as a collection of planes overcomes these problems. This reduces the complexity of the model computation as well as its rendering and yields more photo realistic view synthesis of planar and nearly-planar surfaces. These led to an investigation of planar reconstruction of scenes as seen most recently in [24]. The idea of using planar modeling requires identification

of planes in given images. Layer extraction methods like [25] are used for this purpose. Tracked features are grouped into planes using the layers extracted. We describe a method to estimate plane parameters and camera poses from features tracked from various planes.

1.2 Scene interpretation in Robotics

In this section, we discuss the scene interpretation aspects in robotics specially mobile robotics. The basic goal of mobile robot is to move autonomously through an environment from its current position to some goal position. There are three important tasks that needs to be executed in navigation.

1. The first is the task of *localisation*. Localisation refers to the task of identifying where the robot lies with respect to a pre-defined map or global co-ordinate system. Localisation is performed through an inference process over the robot's representation of the environment and sensor readings from the current location, which is some what *scene recognition*.

2. The second task is that of *planning*, in which the localised robot need to find a path through the environment which leads it to the goal position. The path determined by the robot must be navigable and free from obstacles. Also the path must be optimal in some sense such as time, speed etc., depending on the purpose of the robot.

3. The third task is that of *path execution*, in which the robot generates a sequence of control signals for its actuators, so that robot traverses in the planned path.

The second task can alternatively be used for *map building* which looks at the task of building a 3D map of the environment, which can be used later for navigation. There has also been research on coupling the task of localisation and map building together, which is refered to as Simultaneous localisation and mapping (SLAM) [26]. The part of SLAM research, which uses vision sensors is termed vSLAM [27, 28], and has received much attention. In the context of SLAM, the ability to recognize a visited place is known as the 'loop-closure detection'. It is named as so because the robot needs to perform *scene recognition* at the end of a loop so that the uncertainty linked to its current position will not grow out of bounds. The inability to detect loop closure will mean that the robot is essentially lost. Hence, scene recognition stands as an important step in autonomous robotic navigation.

The power of interpreting the outside world is possible through sensors, which helps the robot to determine what action it should take. Different sensors are used in literature for sensing which include infrared sensors, sonars, lasers, LIDAR, stereo cameras, omnidirectional cameras, monocular cameras etc., The main advantage of using cameras as opposed to other sensors is that they are

extremely cost effective, compact and readily available, they provide a much cheaper mechanism of obtaining accurate 3D information about the world and they are passive sensors. Unlike radar and sonar that have to generate a lot of information first in order to successfully receive information about the environment, vision systems only receive information; they do not transmit any. This passive feature ensures increased levels of portability, durability etc., Unlike other sensor types, vision has the potential to provide rich, semantic information about an environment. Vision provides information regarding the appearance of an environment and objects embedded in it, not just geometric structure or information about the spatial location of objects. Also vision sensors have the information of very far range, where as others have their own limitations. The interest in vision for mobile robotics has been fueled by recent advances in computer vision techniques and the increased capabilities of computing hardware which makes it possible to analyze and interpret images within the time constraints demanded by robotic applications.

Another perspective to view vision sensors based navigation is its use for indoor and outdoor environments. The problem of landmark detection and following have been solved quite successfully in indoor environments. Outdoor navigation is much harder problem compared to indoor navigation mainly due to huge variations in view points and illumination changes. Navigation in outdoor terrains is one of the focus of this thesis.

One of the dreams of an autonomous robotic system is to freely navigate on cluttered and unstructured outdoor environments, specially in Indian context. This involves object detection/avoidance and path planning. The lack of highly structured components in the scene introduces new challenges for autonomous navigation. This navigation system is important because, these systems can be readily employed in military operations and also in civilian applications such as wide-area environment monitoring, disaster recovering, search-and-rescue activities, as well as planetary exploration. Though obstacle detection and avoidance are essential tasks, they are not sufficient for a mobile robot to navigate safely in cross-country environments, because these environments contain several types of terrains such as mud, road, grass, etc., which are hazardous and should be carefully neglected or navigated based on the type of terrain. Hence an effective description of outside world should consist of combination of geometric and terrain type information along with control strategies. Terrain type information extraction is shortly called as terrain recognition/classification [29, 30] in robotics. Terrain recognition enables the robot to navigate safely/intelligently and it also helps the path planner in deciding the optimal path and optimal velocity for traversal.

The problem of terrain recognition can be approached by a combination of various sensing modalities such as 2D and 3D lasers, multiple cameras, vibration sensors [3, 31, 32] or a combination of them [4, 29, 33–35], this thesis explores how much of scene interpretation ability is vested

in a single camera. This investigation is especially crucial since cameras are often less expensive and are not power hungry like laser range finders. Also cameras do provide a rich set of visual features even at longer distance, which helps the robot in better path planning and hence over-coming the problem of "short-sightedness". Recent advances in computer vision [36], machine learning and hardware computing capabilities also motivates us to solve the problem using a single camera.

1.3 Problem statement and Contributions

The goal of this work is to develop solution to some of the problems associated with a robot navigating reliably and effectively using a monocular camera through outdoor urban environments using optimization and machine learning techniques. Towards this end, this thesis presents the following:

1. A robust 3D reconstruction scheme in piece-wise-planar environments using convex optimization techniques is presented. The method is formulated in an L_∞ based Homographic framework that iteratively computes scene plane and camera pose parameters. Existing SVD based method are proposed only for two views and are very sensitive to noise. On the other extreme, iterative non-linear methods like Bundle adjustment are computationally expensive and there is a high chance that they get stuck in local minima. The proposed method handles these issues using popular convex optimization techniques, which are proved to be robust and computationally inexpensive. In a sequence of images, Homographies induced between inter images (if available), which are more accurate and informative are formulated as additional constraints in the framework to arrive at an optimal solution. The method was tested empirically on synthetic data of several random planes and on real data against SVD and Bundle adjustment methods.

2. A fast terrain classification algorithm that allows a robot or vehicle to determine various types of natural terrains using only monocular camera is presented. Most of the existing methods are either limited to ground plane detection or use lasers or IMU for terrain classification. We intend to solve the problem using only monocular camera, without using power hungry and costly hardware such as lasers. We introduce our new dataset for conducting various experiments. The dataset was collected by a monocular camera mounted on top of the vehicle moving in different speeds over 10km in various illumination conditions in urban and rural roads. We empirically study the problem with existing features and classifiers. The best classifier was found to be Random forests. The challenges involved with the existing classifiers is the missing context information. The algorithm handles this issue using a novel partitioning scheme. Various aspects of the algorithm importantly the spatial context, was tested on our

dataset and other publicly available datasets with the existing classifiers.

3. An adaptive terrain classification scheme that allows a robot to determine various natural terrains, where terrains may change their appearance over time gradually is presented. Existing methods are memoryless i.e., they assess the terrain of the captured image without using the previous learned knowledge. Recently, methods which use these memory are being proposed, but these methods require either lasers or stereo for collecting ground truth. The proposed scheme is based on only a monocular camera. The proposed scheme effectively uses the acquired knowledge from previous classification and temporal information. The trained classifier handles the slow drifts in the natural terrains online. The method was tested on our dataset and other publicly available datasets in an experiment, where the vehicle traverses the same path twice.

1.4 Organization of thesis

The remainder of this thesis is organized as follows:

1. In Chapter 2, we give an overview of the basic mathematical concepts related to this thesis. First we introduce homography, and its relation to Camera parameters and pose. Next we briefly describe the SVD based homography decomposition methods. We then introduce the problem of Layer extraction and popular solutions to the problem. We use the Layer extraction methods for segmenting planes described in next chapter 3. Next we give overview of the standard problem in Computer vision the Structure-from-motion and we describe the traditional iterative non-linear optimization method Bundle adjustment. After that, we give brief introduction to convex optimization which is used in Chapter 3. Next we introduce the second major research problem that we deal in this thesis, the terrain recognition. We then briefly describe its applications. After that we give literature review, which includes a brief overview of vibration-based methods, near to far learning methods and few recent methods. We then present a summary of promising color and texture features along with few popular classifiers that are used in literature for our problem.

2. In Chapter 3, we give an overview of the literature in Convex optimization along with its utility. This is followed by sensitivity analysis of the existing SVD methods, which is followed by convex framework for the problem of planar reconstruction. Several experiments, extensions to the framework is described.

3. In Chapter 4, we present our annotated dataset that we use in our experiments. We then present extensive empirical comparisons of various features and state-of-the-art classifiers in

machine learning literature. Next we show how various parameters of the problem affect the classifier performance.

4. In Chapter 5, we extend the Random forest classifier using partitioning scheme, which is followed by several experiments that test the proposed scheme. This is followed by introduction to the novel adaptive algorithm using optical flow. Next we conduct two experiments to test the algorithm. The results show considerable decrease in percentage error compared to Random Forests. Also, the adaptive classifier was able to slowly adapt to appearance changes that occur during the navigation of the vehicle.

5. In Chapter 6, we conclude the thesis. We summarize the contributions of this thesis and comment on limitations and future work.

Chapter 2

Background

2.1 Geometry of Planar Scenes

In this section, we give brief overview of several technical terms and algorithms that we use in the thesis, which are being popularly used in computer vision in the recent years.

2.1.1 Homography

As shown in the Figure 2.1, associating the two images x and x' of a 3D point X becomes impossible without the knowledge of the camera parameters and the value of X itself. However, when the point X lies on a plane Π, a simple geometric entity suffices to map one image point (x) to another (x'). This geometric entity is called the *homography* subtended by plane Π, which is represented by $H_{3\times3}$. Thus in the case of perspective projection, a homography maps one image point x to another x', upto a scale factor.

$$x' = \frac{1}{\lambda} H x \tag{2.1}$$

where λ is the scale factor. Though the homography matrix H has 9 elements, due to scale factor it is parameterized by only 8 parameters. Thus without loss of generality, the last element $H(3,3)$ can be assumed to be unity. Since the above equation is linear, 8 equations are required to solve for the value of H in minimal case, which results in 4 image-to-image correspondences (each correspondence giving 2 equations in x and y image coordinates). In real images, this minimal case is highly sensitive to errors in correspondences, current feature extraction algorithms like SIFT [37] ensure that the homography estimation is quite accurate when the camera poses aren't too far apart. Thus, a RANSAC based approach [1] suffices to weed out incorrect correspondences as they are usually only outliers of the actual function.

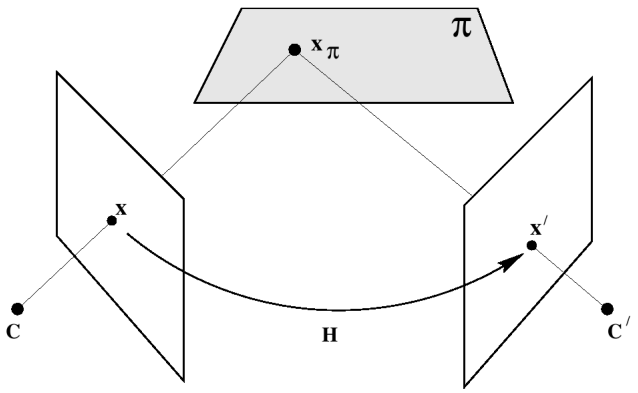


Figure 2.1: Homography induced between two images x and x'. Image courtesy [1]

Homographies are the best suited tools for reconstructing planar scenes, because they directly utilize the perspective mapping of planes and thus stay closer to the original data than methods, which start with point-wise sparse 3D reconstruction [38, 39] and then segment the resulting point cloud into planes. Also, extremely robust solutions exist to compute the homography induced by a plane in two cameras [40].

In this thesis, we use homographies due to (piecewise) planar scenes. The scene planes impose a strong constraint, which has been used mainly for structure and motion recovery. Homographies have several practical applications, for example they are used for mosaicing and super-resolution [38, 41]. If the homography induced by a plane in two images is known, one can find the corresponding features on the images of the plane. This has been used for grouping of coplanar features in wide-baseline settings [42, 43] and for feature matching and also for transfer of features off the plane, with the help of known reference planes and projective invariants [44]. If also borders of the planes in the images are known, they can be used for texture unwraping and for image compression [45]. The problem of motion recovery [46–48] can be linearized by homographies. Measurements on scene planes in perspective distortion is possible through homographies [**?**]. Homographies also allow reconstruction of non-planar scenes, which can be seen as collection of planes and the deviations from these planes, which is termed the "plane-plus-parallax" approach to vision [49, 50].

2.1.2 Homographies and Camera parameters

In this section, we describe the relationship between the homography H relating two images and the relative pose between their corresponding cameras. Let us assume that the two cameras are given by $P_1 = [I|0]$ and $P_2 = [R|t]$. Where I is identity matrix, (R, t) is the relative pose. Let X be the 3D point belonging to the plane represented by $\Pi = [n^T 1]$, and let x and x' be its projections respectively. Then

$$x = P_1 X = [I|0]X \tag{2.2}$$

$$X = [x^T \rho]^T \tag{2.3}$$

Different values of ρ represent different points on the 3D line joining camera center C and 3D point X (Figure 2.1). Thus the value of ρ that satisfies the above Equation 2.3 is $(-n^T x)$. Substituting the value of ρ in the projection equation for the second image, we get

$$x' = P_2 X = [R|t]X = Rx - tn^T x = (R - tn^T)x \tag{2.4}$$

When the internal parameters cannot be assumed to be identity but are known to be different for the two images, the modified equation of the relationship is as follows

$$H = K'(R - tn^T)K^{-1} \tag{2.5}$$

where K and K' are the internal parameters of the two cameras respectively.

2.1.3 Homography Decomposition

Traditional methods for obtaining the camera pose and plane normals from the Homography matrix rely on the Singular Value Decomposition (SVD) of Homography to provide solutions [51, 52]. In both the methods, eigenvalues of either the Homography matrix H or $\mathrm{H}^\top\mathrm{H}$ are used to get upto 8 solutions for $\{\mathbf{R}, \mathbf{t}, \mathbf{n}\}$ and then 6 solutions are weeded out based on many constraints. Finally, the 2 remaining solutions may be disambiguated by either considering a third view or a second plane.

Faugeras SVD-based decomposition Faugeras *et. al* [51] algorithm starts with the singular value decomposition of the Homography matrix, followed by the equation relating the diagonal matrix thus produced to a new set of variables as

$$\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^\top \tag{2.6}$$

$$\mathbf{\Lambda} = \mathbf{R}_{\mathbf{\Lambda}} + \mathbf{t}_{\mathbf{\Lambda}}\mathbf{n}_{\mathbf{\Lambda}}^\top \tag{2.7}$$

Computing the components of the rotation matrix, translation and normal vectors is simple when the matrix being decomposed is a diagonal one. First, $\mathbf{t}_\Lambda$ can be easily eliminated from the three vector equations coming out from Equation (2.7) (one for each column of this matrix equation). Then, imposing that $\mathbf{R}_\Lambda$ is an orthogonal matrix, we can linearly solve for the components of $\mathbf{n_\Lambda}$, from a new set of equations relating only these components with the three singular values (see [51] for the detailed development). As a result of the decomposition algorithm, we can get up to 8 different solutions for the triplets: $\{\mathbf{R_\Lambda}, \mathbf{t_\Lambda}, \mathbf{n_\Lambda}\}$. Then, assuming that the decomposition of matrix $\mathbf{\Lambda}$ is done, in order to compute the final decomposition elements, we just need to use the following expressions:

$$\mathbf{R} = \mathbf{U}\,\mathbf{R_\Lambda}\,\mathbf{V}^\top \tag{2.8}$$

$$\mathbf{t} = \mathbf{U}\,\mathbf{t_\Lambda} \tag{2.9}$$

$$\mathbf{n} = \mathbf{V}\,\mathbf{n_\Lambda} \tag{2.10}$$

Zhang SVD-based decomposition Zhang *et. al* [52] take a different approach by first computing the eigenvalues of $\mathbf{H}^\top\mathbf{H}$, and then using it for further computation of the quantities $\{\mathbf{R}, \mathbf{t}, \mathbf{n}\}$.

$$\mathbf{H}^\top\,\mathbf{H} = \mathbf{V}\,\mathbf{\Lambda^2}\,\mathbf{V}^\top \tag{2.11}$$

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3) \tag{2.12}$$

$$\mathbf{V} = [\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}] \tag{2.13}$$

$$\lambda_1 \geq \lambda_2 = 1 \geq \lambda_3 \tag{2.14}$$

In the first step, values of $\{\mathbf{t}^*, \mathbf{n}\}$ are computed, where $\mathbf{t}^*$ is the normalized translation vector. Subsequently, the rotation matrix is obtained as $\mathbf{R} = \mathbf{H}\,(\mathbf{I} + \mathbf{t}^*\,\mathbf{n}^\top)^{-1}$. Eight solutions are obtained in the following manner

$$\mathbf{t}^* = \pm\frac{\mathbf{v'_1} \pm \mathbf{v'_3}}{\zeta_1 \pm \zeta_3} \tag{2.15}$$

$$\mathbf{n} = \pm\frac{\zeta_1\mathbf{v'_1} \pm \zeta_3\mathbf{v'_3}}{\zeta_1 \pm \zeta_3} \tag{2.16}$$

where equations in numerators and denominators share the same sign in all variations. The variables $\{\mathbf{v'_1}, \mathbf{v'_3}, \zeta_1, \zeta_3\}$, are functions of the eigenvalues $\mathbf{\Lambda}$ [52].

2.1.4 SFM and 3D reconstruction

Humans are naturally able to infer the location and structure in three dimentional world, using only two dimensional images perceived through eyes. This process of inferencing depth from images is seemingly an effortless task, but it is very hard to implement in a computer. The task of recovering the 3D structure of the scene and sensor motion from a set of 2D image frames obtained from an optical camera refers to Structure from motion(SfM). SfM is used in various practical applications, which include 3D model reconstruction, 3D motion matching, camera caliberation, perceptual computer interfaces, robotics, image mosaicing, etc.

Solutions to the problem of SfM may be broadly divided into corresponded SfM and correspondence-less SfM. Corresponded SfM requires some kind of features to be tracked, where as Correspondence-less SfM is generally based on phase component [53] of Gabor transforms of images, where the phase difference of the gabor images is inversely proportional to the depth of the scene. In this thesis, we deal with corresponded SfM and we refer it with SfM. There are two main assumptions that are inherit for the task of SfM.(i) The scene is static i.e., the objects are rigid. and (ii) There exists some method to extract a set of 2D features from images. 2D features may be points, lines, curves, etc or combination of them. It is assumed that these 2D features are detected and associated to their corresponding features in the available images. These 2D measurements stand as the inputs to the problem of SfM.

SfM is an active research area from almost 30 years. Unfortunately the current literature is still far away to what human can perceive. Its a hard problem and of interest to both computer vision and AI communities. Multiple approaches have been proposed in literature [54–58]. These range from perspective to orthographic, 2-frame or stereo to videos, linear(SVD) to non-linear(Optimization based methods) etc., Each method has its own advantages and disadvantages with different input features, different accuracies and different abilities. The choice of the framework depends on the application that we are interested in. In this thesis, our application of interest is 3D reconstruction specifically in piece-wise-planar environments, where we use homographies for obtaining dense reconstructions avoiding point-based reconstructions.

In the following we briefly describe the typical solution for SfM.

- 2D features(points or lines or curves or etc.,) are detected and associated.
- A projective frame among the available views is initialized as the reference frame.
- Projective camera matrices are chosen which satisfy the computed Fundamental matrix from correspondences.
- Initial solution for the structure of the scene is obtained.

- Results are refined using bundle adjustment methods. [59].

2.1.5 Bundle Adjustment

Bundle adjustment [59] is a standard iterative non-linear optimization technique , which uses Levenberg-Marquardt internally. Bundle adjustment needs initialization. This initialization is used to minimize the following error over the normals and the translations

$$(R, t, n_j, d_j) = \arg \min_{{}^kR, {}^kt, n_j, d_j} \Sigma_k \Sigma_j \Sigma_i [\frac{h_i}{h_9} - \frac{x^T A_i x}{\overline{x}^T A_9 \overline{x}}]^2 \tag{2.17}$$

where, $x = ({}^1R^s, \ldots, {}^KR^s, {}^1t^T, \ldots, {}^Kt^T, n_1^T, \ldots, n_J^T, d_1, \ldots, d_J)$ and A_i is a matrix s.t. $x^T A_i x = g_i$ and $\overline{x}$ is x with the initial SVD estimates of ${}^kR, {}^kt, n_j, d_j$ substituted. The main disadvantage of this technique is that they are computationally demanding and one might end up getting local optimal solution.

2.2 Layer extraction

Layer extraction in videos is essentially segmenting or representing the images into some number of *sub-images*(See Figure 2.2), in such a way that pixels within each sub-image share some common 2D parametric transformation. Layer extraction is an initial step in most of the problems related to the video processing. In the following we give some examples:

- In scene reconstruction, one can attain dense reconstructions by using layer representation based SFM, avoiding sparse reconstructions which are based on feature points.
- In motion analysis [60–62], the hardest problem of finding occlusion relationship is explicitly a layer extraction problem. Image motion estimation is inherently an ill-posed problem [63] due to the aperture problem, in order to estimate the motion, it requires additional smoothness constraints such as parametric model that assumes some pixels share a common model with a few parameters [64] or regularization [65, 66]. However, it is not necessary to apply such constraint across motion boundaries, which are not known prior to the motion estimation. In layer representation, we can enforce such smoothness constraint only inside each layer, and explicitly represent the non-smoothness at the boundaries among layers.
- In visual navigation, layer representation can be used to extract and represent the ground layers (roads, terrains), and objects (cars, pedestrians, etc). Ground layer is useful for obstacle detection in robotics and estimating the car ego-motion [67].

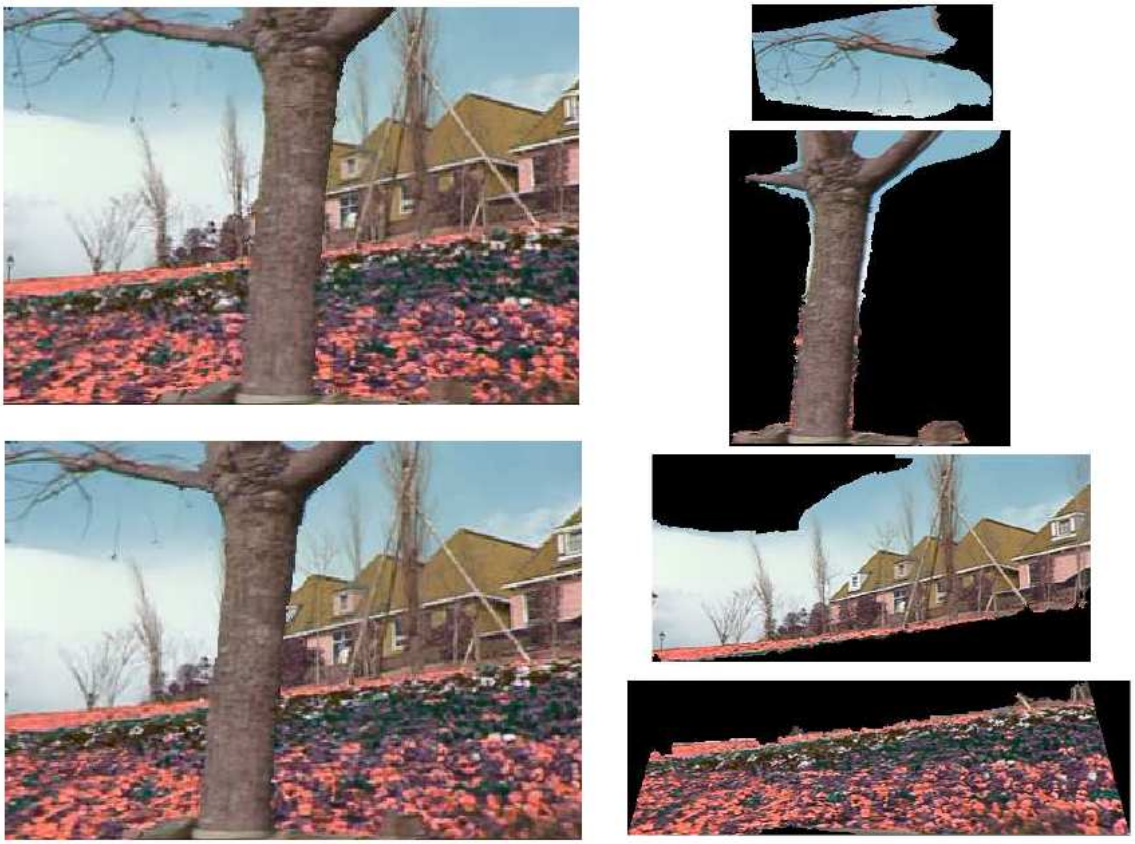


Figure 2.2: Left: Consecutive frames from a garden sequence, Right: Sub-images or layers in the video. Image courtesy [2]

- In object detection and recognition, layer representation gives the first cut solution to detecting several objects. For instance consider a video in which a dog runs parallel to the motion of the camera from left to right, in layer representation the dog is one of the layers.

Layer extraction problem has three major issues which are (i) Segmentation (what region of image belongs to one layer ?) (ii) Motion (What motion does the camera under went ?) and (iii) Number of layers (How many number of layers are present in the video ?). These three issues are coupled problems, i.e., On one hand, spatial layer supports (including number of layers) are required to estimate the motion model for each layer. On the other hand, assigning pixels to layers requires the knowledge of layer motion model.

In the following we briefly summarize few popular approaches to layer extraction.

EM approach : A natural approach to solve the coupled problems in layer extraction is the Expectation Maximization (EM) algorithm [61, 68–71]. In such an approach, the likelihood of the video data is formulated as some mixture model, such as the mixture of Gaussians, with each mixture component representing a layer. In EM approach, there will be an iterative E-step and M-step, and then MDL principle [72] is used to find the number of layers in the video, this was modelled as a search problem in [68],which is a costly operation. Initializing(for example [68]) the number of models and the motion for each model is an important but difficult step for EM approach [71, 73]. Without good initialization, EM algorithm may not converge to desired optimal solutions.

Dominant approach : This approach is one of the top-down approaches for layer extraction problem. This approach assumes that there is always a dominant layer in the given sequence of images. The approach consists of several iterations. In each iteration, the current dominant layer is extracted using dominant motion estimation [74–76] using robust estimator [77, 78]. After that, the detected dominant layer is segmented out, and the whole process is repeated on the remaining portion of the image until there is only one layer in the image or all the pixels in the image are assigned layers. The main drawback of this approach is the very existence of the dominant layer, which might be always present.

Grouping approach : The grouping approach was introduced to overcome the problems of dominant approach. Grouping approach is a bottom-up approach. It is based on the fact that the 2D homography of a computed from several regions of the plane remains the same(upto a scale factor). In this approach, the image is first divided into small blocks (say 16×16), and the 2D homograpy is computed between the reference frame and the other frames. Here we want to extract layers in the reference frame. After that the 2D homographies are clustered using popular clustering methods such as k-means [79] and normalized graph cut [73]. Each cluster represents a unique layer in the image. Blocks corresponding to a cluster are grouped and declared as one layer. The main

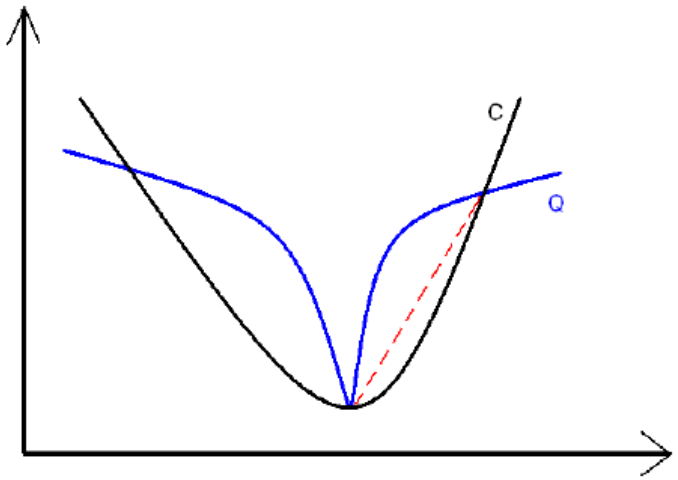


Figure 2.3: A quasiconvex function Q on $\mathcal{R}$ which is not convex. It is plotted with a convex function em C. However, any horizontal line slices Q in atmost two points, thus creating only convex sublevel sets.

advantage of this approach is that it does not require the number of layers as input and also it doesn't assume anything about the given sequence. However, grouping purely based on local measurements is highly noisy and it also ignores the global spatial and temporal constraints.

Subspace approach : Subspace approach [2] is an advanced approach to grouping approach, overcoming the problems with the grouping approach. The important issue of grouping in high dimensional space is handled by the introducing of new subspace which is smaller and hence one can easily perform clustering. The main problem with the grouping approach is the missing global spatial-temporal constraints. Subspace approach enforces such constraints by computing a subspace from homographies intelligently. In this approach, a measurement matrix is constructed by stacking up the relative affine homographies of small image blocks, and then the measurement matrix is decomposed using SVD to calculate a subspace of size 4 or less. Such a low dimensional subspace is possible because it is the measurement matrix is inherently rank deficit. This approach also provides a constraint to detect outliers in the local measurements, which makes the layer extraction robust. However, the subspace computation (factorization of measurement matrix) is a non-linear objective function, which may get stuck in local minima.

2.3 Convex Optimization

A function $f : \mathcal{R}^n \longrightarrow \mathcal{R}$ is *convex* if $\mathbf{dom}\ f$ is a convex set and if for all $x, y \in \mathbf{dom}\ f$, and with $0 \leq \theta \leq 1$, we have

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y) \tag{2.18}$$

where a set C is *convex* if for any $x_1, x_2 \in C$ and any θ with $0 \leq \theta \leq 1$, we have $\theta x_1 + (1-\theta)x_2 \in C$. Figure 2.3 shows typical examples of convex and quasiconvex functions.

A function is defined as quasiconvex [80], if the domain on which the function attains any value less than a given threshold α, is a convex set, for any arbitrary value of α. Such a set is called a sublevel set corresponding to the value of α. Some functions, like the linear fractional function $\frac{ax+by+cz}{dx+ey+fz}$ where (x, y, z) are variables, are known to be quasi-linear (both quasiconvex and quasi-concave) under certain conditions (denominator > 0). As can be seen, many functions like the perspective projection function for a pin-hole camera and the point transfer function using Homographies, can be modeled as a linear fractional in the variables representing the camera matrix and the Homography matrix respectively.

Quasi-convex functions are minimized using what is called the *bisection method*, an iterative algorithm which solves the problem by finding the smallest sublevel set that contains the global minima of the quasiconvex function. This is done by solving a set of *convex feasibility problems*, one in each iteration. If p^* is the optimal value of a convex function $f : \mathcal{R}^n \longrightarrow \mathcal{R}$, then define $\phi_t : \mathcal{R}^n \longrightarrow \mathcal{R}, t \in \mathcal{R}$ as

$$f(x) \leq t \Longleftrightarrow \phi_t(x) \leq 0$$

such that $\phi_s(x) \leq \phi_t(x)$ whenever $s \geq t$. Then the *bisection method* solves the following feasibility problem at each iteration

$$\begin{aligned} \text{find} \quad & x \\ \text{subject to} \quad & \phi_t(x) \leq 0 \\ & \text{other constraints} \end{aligned} \tag{2.19}$$

If the above problem is feasible then we have $p^* \leq t$, and conversely infeasibility denotes $p^* \geq t$. The *bisection method* maintains an upper and a lower bound for p^*, based on the above feasibility problem. At every iteration, this bound is halved by changing one of the two bounds. Convergence happens when the difference between bounds is sufficiently small.

In order to apply the *bisection method* to problems in MVG, we need to first prove that the underlying objective function is quasiconvex. Although functions like the linear fractional function is proved to be quasiconvex, typical objective functions in MVG involve minimization a geometric

error of the form

$$d \quad = \quad (y_1 - f(x) * y_2)^2 \tag{2.20}$$

where (y_1, y_2) are typically correspondences (2D or 3D points or both), and $f(x) * y_2$ is a linear fractional function function whose parameter x needs to be determined such that d is minimized. Such a formulation requires the following two concepts to prove quasiconvexity (repeated from [81] for completeness):

1. If $f_1(x), \ldots, f_m(x)$ are quasiconvex functions, then $\max_i f_i(x)$ is also quasiconvex.
2. Let $f_i(x), i = 1, \ldots, m$ be affine functions, *i.e.*, $f_i(x) = a_i^\top x + b_i$. Then

$$\frac{f_1(x)^2 + \ldots + f_{m-1}(x)^2}{f_m(x)^2}$$

with domain $\{x \mid f_m(x) > 0\}$ is quasiconvex.

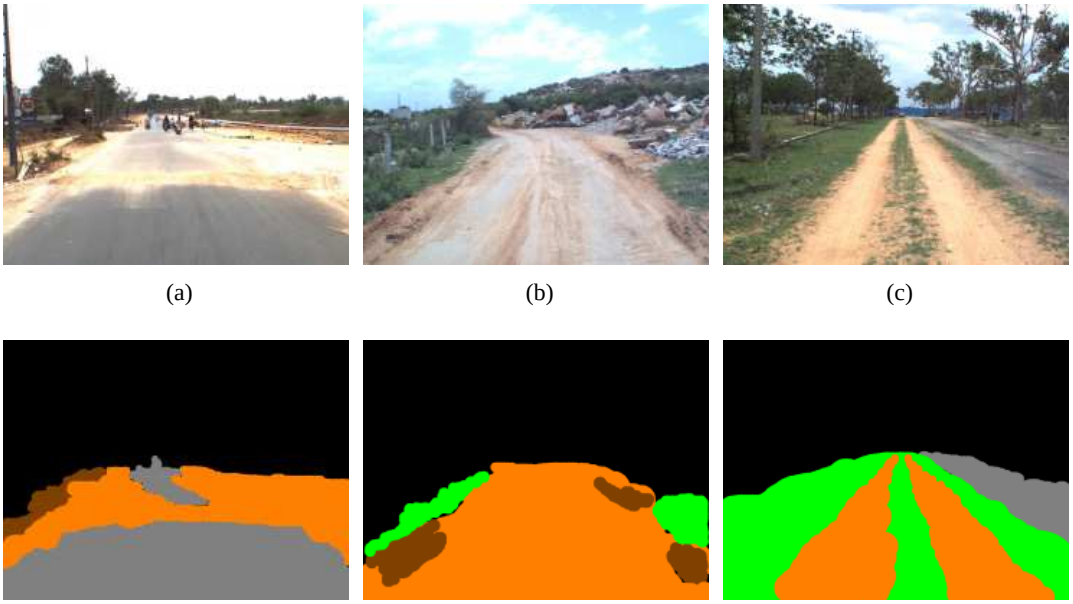
2.4 Terrain Classification

In mobile robotics, much of the interest has gone in understanding scenes containing rural and Urban terrains for many robotic tasks such as navigation and planning. The goal of terrain classification [30, 82] is to recognize various terrains that occur in urban and rural environments in an automated fashion. An automated solution to the terrain classification is very crucial in various domains such as (i) advanced driver assistance systems [83], (ii) autonomous navigation, (iii) remote sensing, (iv) urban and rural planning. Figure 2.4 shows the some of the sample images and their respective desired output.

For instance a mobile robot navigating outdoors comes across various terrains such as soft and slippery terrains, hard and smooth terrains or rocky and undulating ones. The navigation strategy for the robot differs greatly based on the kind of terrain it traverses, the limits on its velocities vary according to these surfaces. An algorithm capable of prior judgment of the terrain provides the well needed time for the robot to adapt its velocity planner and thus becomes a vital cog in outdoor navigation systems. While in this thesis we focus on the problem of classifying terrains for autonomous outdoor navigation, the broader scope of the problem is indeed evident. For example one can make use of such algorithms in driver assistance and there by ensuring safety.

2.4.1 Literature review

One way to determine the terrain type is to directly estimate terrain parameters like cohesion or slippage from sensor measurements. Another way is to group the terrain into classes like asphalt,



(a) (b) (c)

(d) (e) (f)

Figure 2.4: First row: Sample frames from our dataset. Second row: Desired output

dirt or gravel, and to learn these classes from training examples. Once the robot has learned the different classes, it can classify new terrain data according to the learned model.

Various methods have been proposed in literature for the problem of terrain recognition. They can be broadly divided into ladar-based methods [84] (which use laser, radar etc.,), vibration-based methods [85] (which use accelerometers, IMU etc.,) and cameras based methods [29, 86, 87]. Ladar-based methods usually fit a plane on the obtained ladar data for recognizing terrain. They often focus on segmenting the ground surface from vegetation or different kinds of obstacles (e.g. rocks) instead of estimating the type of the ground surface itself. Other ladar based methods divide the ground surface into navigable and non-navigable regions [88].

Vibration based methods

Among vibration-based methods, usually accelerometers are used to measure the vibration perpendicular to the motion of the vehicle. The raw measurements of the accelerometers are generally very similar for different types of terrain (See Figure 2.5). Thus it is beneficial to transform these data to a more significant representation. In [3], several representations are compared, among them the popular ones are Fast Fourier Transform (FFT) representation as suggested by Sadhukhan [31], a log-scaled power spectral density (PSD) as used by Brooks and Iagnemma [32], and a more com-

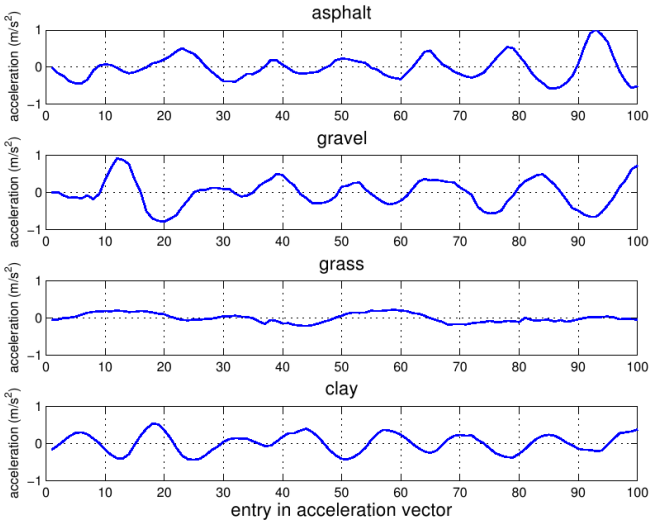


Figure 2.5: Sample raw acceleration data for various different types. Observe that except for grass other terrain record very similar measurements. Image courtesy Weiss *et. al.* [3]

pact representation based on simple features calculated from the acceleration vector like number of sign changes. In [89] and [32], Brooks and Iagnemma transform their acceleration data to a power spectral density (PSD) representation. A log-scaling of the magnitude reduces the dominating effect of high-magnitude frequency components. Then, they used the principal component analysis (PCA) to reduce the dimensionality of their feature vectors and to separate the signal from noise. To separate feature vectors of different classes, they use linear discriminant analysis (LDA). They train a set of pairwise classifier, one classifier for each possible pair of terrain types. These classifiers take into account both the distribution of feature vectors within a single class as well as the separation of the class means, and compute a discrimination vector, then they use Mahalanobis distance as their distance metric. Though these methods are highly reliable and are independent of environment and climatic illumination conditions, the terrain can be classified only while the robot traverses it, but not beforehand.

Near to Far learning methods using lasers or Stereo cameras

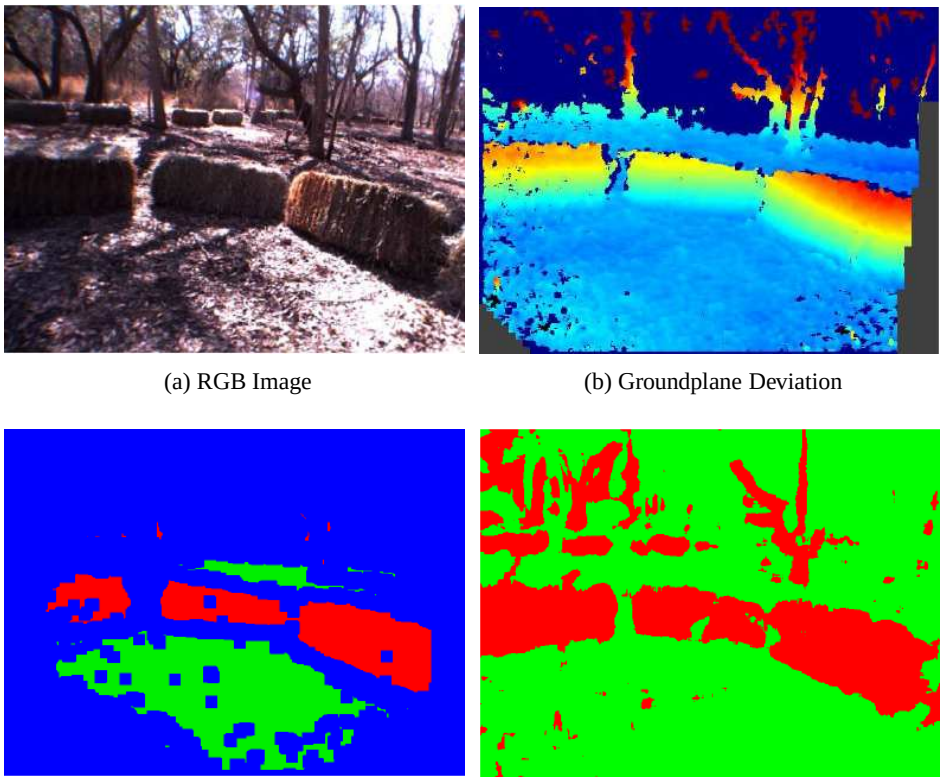
There are methods that use a combination of laser and images or stereo based data for purpose of annotation or ground plane extraction and training [4, 29, 33–35]. These address the problem of classifying the terrain into navigable and non navigable sections for further use by a planner module. These methods follow a canonical form of using camera along with lasers or stereo-rig,

i.e., as the robot navigates through the terrain, dense 3D data is acquired using lasers or stereo cameras. A groundplane model is fit and subtracted out, resulting in an estimate of groundplane deviation Figure 2.6b. Near-field labels from both the groundplane and obstacle classes are extracted according to small and large groundplane deviation values, respectively Figure 2.6c. The near-field stereo labels are sampled to create a balanced training set, features are extracted from the image at these sampled points, and a machine learning model is trained on the resulting training data. Finally the classifier is evaluated over the remainder of the image, including the far field, to arrive at a final terrain classification Figure 2.6d.

Recent literature

Among the recent literature, we surveyed the work reported in [86] on monocular terrain classification comes closest to ours. Dima *et. al* [90] trains separate classifiers on data from laser, infra-red camera and monocular camera and uses AdaBoost to combine the output. Bradley *et. al* [91] uses multi-spectral camera to detect chlorophyll content for recognizing grass and trees. Recently, Blas *et. al* [33] uses pre-segmentation algorithm based on clustering using LBP features before training phase, Vernaza *et. al* [30] uses Markov random fields framework for training on set of their own training data and report accuracy in the range 68%-88% on four datasets. Procopio *et. al* [4] adds memory to the machine learning model by using ensemble of classifiers, they report an accuracy of around 90% on their own publicly available datasets, but they consider only two classes they are traversible vs. non-traversible path.

While the problem can be approached by a combination of various sensing modalities such as 2D and 3D lasers, multiple cameras or a combination of them, this thesis explores how much of scene interpretation ability is vested in a single camera and is thus different from methods that use multiple sensing modalities such as those cited above. This investigation is especially crucial since cameras do provide a rich set of visual features even at longer distance, which helps the robot in better path planning and hence over-coming the problem of “short-sightedness”. Often mobile robots are equipped with limited power systems, it is often desirable to use low power consuming sensors like monocular camera rather than high power consuming sensors such as lasers. Also, cameras are much cheaper compared to ladars or vibration sensors. These factors motivates the use of monocular camera to perform terrain classification. As a part of a larger effort of terrain evaluation by single camera, we manually annotate the data offline. We use these annotated images for automated evaluation. Unlike many previous approaches, which deals with the problem of detection of navigable region, we deal with the complex variant of the problem, which is about classifying the terrain ahead into commonly observed scenarios.



(a) RGB Image (b) Groundplane Deviation

(c) Near-field Labels (d) Final Classification

Figure 2.6: Demonstration of near-far learning using stereo-camera or lasers to obtain dense data. Image courtesy Michael *et al. [4]*

2.4.2 Features

Color features

For any learning based method selecting meaningful features for the classification task is very important. Color cue has been used in literature in various forms, such as color histogram [34], [29], [4] average red and average R+G [86], HSI color space [83] etc., Recently, Carlos *et. al* [92] use U,V components in the LUV color space and report better performance. As representative set of features based on color, we use three features, they are histogram of R,G and B components in the RGB space, histogram of H,S and I components in HSI space and histogram of L,U and V components of LUV space. We quantize each component to 60 bins, hence the size of histogram of color in any space will be 180.

Texture features

As second class of features, we choose is texture. Different types of texture features have been proposed in literature. Many of them are based on using filters such as multichannel filtering [93], LM filter banks [94] etc., We use LM filter banks (see Figure 2.7a) as our first representative features for texture. For calculating the histogram of LM features for a image block, we use a similar approach used in [95], where we take histogram of maximal response filter indices's along with mean and variance of the maximal filter, by which we have a feature vector of size 52.

Linear binary pattern feature (LBP) (see Figure 2.7b) is a gray-scale invariant texture primitive statistic. For each pixel in the neighborhood of the pixel, a binary code is produced by thresholding with the center pixel. A histogram is created to collect up the occurrences of different binary patterns. A related work [96] on recognizing real-world textures was proposed, in that they experiment with different LBP based features and report good classification performance. Recently Blas *et. al* [33] used LBP based feature for segmenting the image as the first step for the problem of terrain classification. In our experiments we use basic uniform LBP feature as our second texture feature.

Recently texton based features were used in [86], texton based representation considers a texture as union of features with specific appearances, without regard to their location [97] and they report that textons alone can classify the terrain with high accuracy. However textons are based on "bag of words" features, which makes them computationally very costly. Therefore we limit our attention to LM filter banks and LBP histograms only.

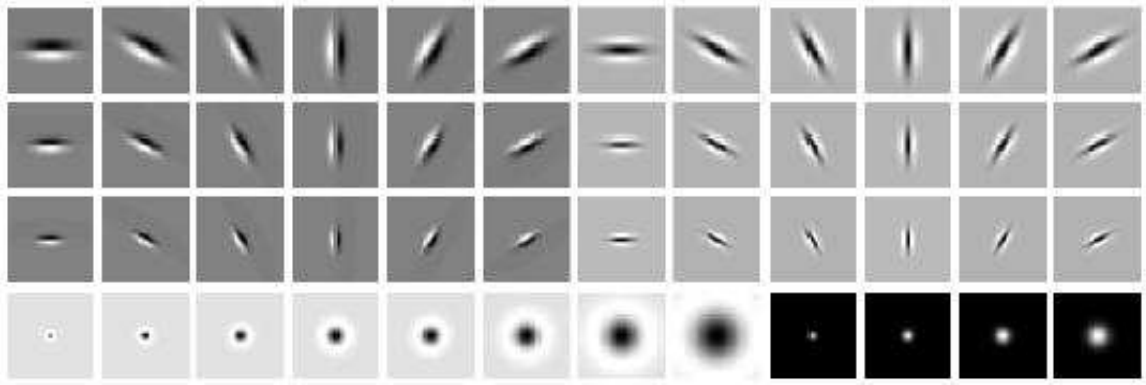
2.4.3 Classifiers

Performance of color, texture and combined descriptors are evaluated on a set of popular and promising classifiers.

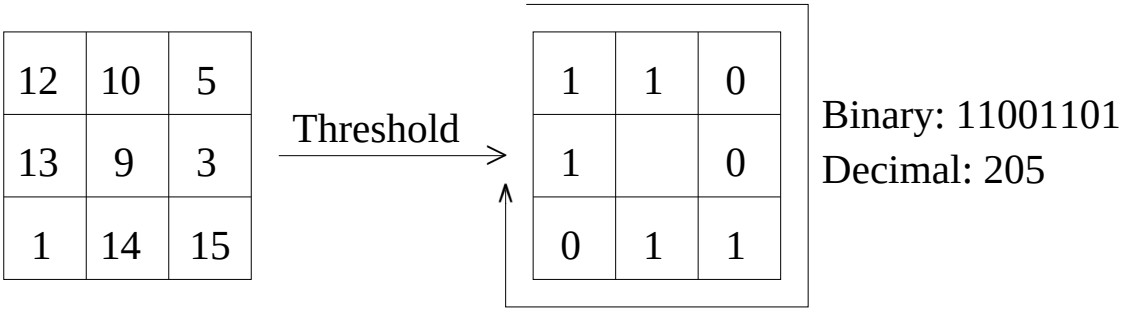
Naïve Bayes

Naïve Bayes classifier is a popular but simple classifier with strong independence assumptions within the features and is based on Bayes reasoning. It has the main advantage of being able to handle a large number of features. This classifier is known to be mathematically optimal under restricted settings.

Let X be a vector whose class label is unknown. Let A be some given hypothesis, such as "vector X belongs to a specified class C". For performing classification, we need to find the conditional probability $P(A|X)$ – the probability that the hypothesis A holds, given the observed vector X. $P(A|X)$ is called posterior probability of A conditioned on X. In contrast, $P(A)$ is the prior



(a)



(b)

Figure 2.7: (a) LM filters has a total of 48 filters, which are 2 Gaussian derivative filters at 6 orientations and 3 scales, 8 Laplacian of Gaussian filters and 4 Gaussian filters. (b) Basic Linear Binary Pattern operator

probability. The posterior probability, $P(A|X)$ is based on more information (such as background knowledge) than the prior probability, $P(A)$, which is independent of X.

Similarly, $P(X|A)$ is posterior probability of X conditioned on A. $P(X)$ is the prior probability of X. Bayes theorem provides a useful way of calculating the posterior probability, $P(A|X)$, from $P(A)$, $P(X)$, and $P(X|A)$. Bayes theorem can be stated as:

$$P(A|X) = \frac{P(X|A)P(A)}{P(X)} \tag{2.21}$$

K-Nearest Neighbor

The k-nearest-neighbor (K-NN) algorithm is one of the simplest machine learning algorithms. However, it often performs very well and therefore, it is an important benchmark method. This method classifies samples based on the closest training samples in the chosen feature space. Given a test sample, it selects the closest k training samples in the training set and reports the dominating label among the closest k training samples. If there is a draw, simply the label of the closest sample is chosen as the label of the test sample. Generally the choice for k should be an odd number. In experiments, selecting k among the values $k \in \{1, 3, 5, 9, 11, 13\}$ is sufficient. The popular distance measures used to find the nearest neighbor are Euclidean distance, Mahalanobis distance, City block (Manhattan) distance, Chebyshev distance, Minkowski distance, Canberra distance, Bray Curtis distance etc.

Artificial Neural Network

Artificial Neural Network (ANN) classifier tries to simulate the structural and functional aspects of biological neural networks. Artificial Neural Network (ANN) classifier are used to model complex non-linear relationships in data. There are two types of learning modes for ANN's, they are batch mode learning and sequential mode learning, In batch mode all the training samples are used at once to update the parameters in the objective function, this mode requires huge amounts of memory to train, where as in sequential mode learning, the parameters of the objective function are updated by learning from a single training sample. If one has to train on huge amount of data, sequential mode is the natural choise of training. Though ANN's takes huge time in the training phase, the testing phase is much faster compared to other classifiers such as K-NN.

SVM's

Support vector machines(SVMs) have become highly popular classifiers in the recent past. SVM's are large margin classifiers with high generalization capability [98]. Initially, SVMs are designed

for binary classification task assuming the data is linearly separable, SVM constructs a optimal hyperplane in the input feature space, by maximizing the margin (distance) between two parallel hyperplanes which are constructed on each side of the separating hyperplane. Among the popular variants of linear multiclass SVM classifiers, we choose 1 vs 1 multiclass classifier, where pair-wise classifiers are created, and at the classification step, the majority of all the classifiers is chosen as the final result, which we call as SVM-L. For handling data, which is not linearly separable, SVMs are extended by using Kernel trick.

Kernel trick transforms the input feature space to higher dimensional space, which allows SVM's to fit the maximum-margin hyperplane in the transformed feature space, which relies on basic assumption that non-linear data may be linearly separable in higher-dimensional space. There are different types of kernels(K) available in literature, such as Radial-basis kernel, intersection kernel, laplasian kernel, polynomial kernel etc., In our experiments we use popular Radial-basis function (RBF) kernel among various available kernels, which we call SVM-K. Training an SVM requires solving the following quadratic optimization problem:

Maximize:

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{2.22}$$

subject to constraints $\alpha_i \geq 0, i = 1, 2, \ldots, l$, and $\sum_{i=1}^{l} \alpha_i y_i = 0$ where α_i are the Lagrangian multipliers corresponding to each of the training data points x_i.

The decision function is given by:

$$f(x) = sgn(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x)) \tag{2.23}$$

where K is the kernel function.

Random Forests

Random forests (RF) (see Figure 2.8) is a classification algorithm that uses an ensemble of unpruned decision trees, each of which is built on a bootstrap sample of the training data using a randomly selected subset of feature space dimensions [99]. The final output is the mode of class's output by individual trees. We use an implementation of the RF available in the matlab environment [100]. This implementation is based on the original Fortran code authored by Leo Breiman, the inventor of RFs.

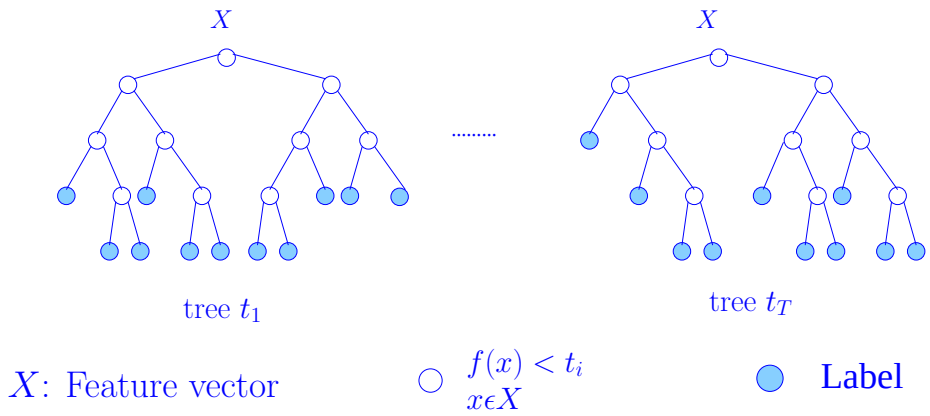


Figure 2.8: Random Forests, containing two types of nodes in every tree, at each un-filled node, a decision function $f(x)$ is defined on random subspace $x \in X$, where X is feature vector, and the filled node is the class label. Majority voted label from all the trees is the final label of the random forest.

Chapter 3

Piece-wise planar scene reconstruction

3.1 Introduction

Convex optimization methods have achieved success in the estimation of various geometric quantities like homography, pose, 3D point cloud (triangulation) [40,81] etc. One of the reasons that make convex optimization an attractive choice for geometric problems is its ability to produce accurate results even with noisy data. Owing to this property, they can be used to handle cases with considerable noise where most other methods often perform poorly. On the other end, these methods have algorithms that are fast enough to be used for real world applications [80]. Moreover, modeling a given problem in a convex framework could lead to a certificate on the optimality. Particularly for cases where the objective function is either convex or quasi-convex, there exists only a single global minima. A theoretical guarantee reinforces our confidence in the estimates derived through a convex framework. Lack of such theoretical guarantee is an issue of common occurrence with most other optimization frameworks that suffer from the trap of local minima. They rely heavily on the quality of the initialization used to run the optimization. Such inability to comment on the optimality hinders the reliability of the method and its estimates. Increasing complexity of objective functions further adds to the unreliability of these algorithms leaving them unusable for practical use. Such complex surfaces and manifolds are cases of common occurrence in computer vision. This stresses the need to reformulate the problems in a convex framework.

We approach the problem of reconstruction in piece-wise-planar scene using convex optimization techniques. We describe a method to estimate plane parameters and camera poses from features tracked from various planes in a given video.

3.1.1 Contributions

In this chapter, we make the following contributions.

- We introduce objective functions for producing optimal estimates of pose and plane parameters, along the lines of [40].
- Since the L_∞ norm is known to be sensitive to outliers, we show how adding extra constraints can increase the robustness of our algorithm.
- We show how a Branch and Bound(BnB) algorithm may be formulated for the computation of optimal rotation between views [101].
- This work was published in *Asian Conference on Computer Vision in 2009* [102]

3.1.2 Organization

The rest of this chapter is organized in the following manner. Section 3.3 sets the problem of pose estimation in a homographic framework and motivates the need for the use of optimization. Section 3.4 presents our solution and algorithm details. Experimental analysis on synthetic and real-world sequences are done in Section 3.5 and finally, we summarize with a discussion on future directions and applications in Sections 3.6.

In this chapter, we explore the use of the property of convex optimization for piecewise planar reconstruction. We show that when the problem of 3D reconstruction is posed as the computation of camera pose and scene plane parameters, the resulting objective functions are quasiconvex or convex in nature, and have good resilience to noise.

Owing to this property, they can be a useful "bridge" between SVD based initialization methods like Factorization that are sensitive to noise and the accurate results replacing non-linear minimization methods like Bundle Adjustment that require good initialization.

Also while computation of robust Fundamental Matrices [103] has been a tricky issue, homographies are comparatively simpler to compute accurately. The section 2.3 explores background on convex optimization for building the necessary notation that will followed in the rest of the chapter.

3.2 Technical Background

Planar Reconstruction Homographies, like fundamental matrices, can also be expressed as a function of the camera pose, and can be decomposed using SVD in a similar manner [52, 101]. Given that now algorithms for automatically 'recognizing' planes in a video exist [104], a robust

homographic framework for using planar models is worth exploring. The reconstruction of a scene can be viewed as a two step process, where camera poses are estimated first, and 3D quantities next. The estimation of the camera pose from image sequences consists of optimizing a six parameter vector $p = \begin{bmatrix} \alpha_x & \alpha_y & \alpha_z & t_x & t_y & t_z \end{bmatrix}$ for every frame , where rotation and translation are parametrized by three parameters each. Recently, globally optimal solutions to pose [101] have been proposed, that use Second Order Cone Programming (SOCP) to estimate pose given point correspondences. The next phase is computation of 3D geometry. For planes, this corresponds to the optimization of four parameters $\begin{bmatrix} n^\top & d \end{bmatrix}$ where n represents the normal, and d the perpendicular distance from world origin. Optimizing over these parameters is relatively less well researched in the literature as opposed to triangulation for point clouds.

Some of the recently introduced quasi-convex objective functions for estimating quantities like homography form the inspiration for our approach [81]. We also adopt the L_∞ framework, motivated by its ability to handle large amounts of data while being able to provide quick solutions to optimization problems [81, 105].

On the application front, some of the closest works are related to 3D tracking [106] and projective Bundle Adjustment (BA) [107]. Similarity to the tracking work is limited to our motivation to propose SOCP related objective functions. A more closely related work is projective BA, where an iterative technique is proposed, that performs camera resectioning and triangulation to recover structure and pose. However, we differ significantly in our approach and our objective functions. Another related work is Bundle Adjustment with constraints [108]. Again, we differ in that we compute the reconstruction from homographies directly, rather than using them to impose constraints on the geometry of 3D points.

Recent study of bi-linear problems in computer vision has relevance to our work [109], since the relation between a homography and plane and pose parameters is essentially a bi-linear one, with terms involving (R, d) (rotation,plane perpendicular distance) and (t, n) (translation, plane normal). However, the formulation proposed in [109] requires that the entire set of plane and pose parameters need to be optimized together. Estimation of rotation parameters becomes infeasible in such a scenario. Thus we do not resort to a formulation along the lines of [109].

The conditions of orthonormality of rotation matrix are troublesome for the problem of pose estimation. The non-convexity of these constraints suggests the use of under-estimators. Since algorithms for this purpose already exist [101], in our experiments, we have set rotation to be constant and only minimized for the remaining parameters (t, n, d), while treating the issue of rotation in a separately. Our experiments with initialization accuracies (Figure 3.1a), show that SVD decompositions produce better estimates for rotation in the presence of noise, as compared to translations and normals. We propose a formulation along the lines of [101] that may be used to optimize rotation,

while keeping the essential structure of our solution, the same.

3.3 Homographic Framework for Planar Reconstruction

Homographies can be decomposed to estimate camera poses and plane parameters using singular value decomposition (SVD) technique. SVD techniques are known to be sensitive to noise [?]. Further more SVD techniques cannot be used exploit information from multiple planes and views to make a more reliable and consistent estimate. Such shortcoming makes SVD techniques unfit for large scale applications where images of multiple planes across multiple views are available. This stresses the need for a unified framework that can make reliable estimates consistent with the data and robust to noise from a configuration multiple frames and images. In the following section we analyze performance of various SVD based techniques,their implementation issues and their resilience to noise.

3.3.1 SVD based Techniques

Let there be m planes in the world, characterized by the parameters $\left[n^1, d^1, \ldots, n^m, d^m\right]$. The jth plane is characterized by the parameters (n^j, d^j), where n^j represents the normal of the plane and d^j represents the perpendicular distance from world origin. Let there be two cameras with external parameters $[\mathbf{I} \mid \mathbf{0}]$ and $[\mathbf{R} \mid \mathbf{t}]$. For simplicity, let us assume that the internal parameters of the cameras are set to identity ($\mathbf{K} = \mathbf{I}$). Thus the Homography induced by the jth plane between the two views is given by

$$\mathrm{H^j} = \left[\mathbf{R} \; - \; \frac{\mathbf{t}n^j}{d^j} \right] \tag{3.1}$$

Decomposition algorithms for obtaining camera pose and plane normals from homography matrix using Equation 3.1 are well known [51, 52]. However, since, the process of pose computation from correspondences through the homography matrix involves two SVDs, a theoretical sensitivity analysis of such algorithms is difficult and approximate [?]. Thus it is more advantageous to do an empirical study of the error in the estimation of plane and pose parameters, given noise in image correspondences.

Figures(3.1a-3.1c), depict the poor performance of one of the SVD based decomposition algorithms [52]. The experiments consisted of adding increasing amounts of noise to a previously determined set of normalized image correspondences. Homographies obtained after a standard RANSAC routine were then decomposed to obtain estimates of the plane and pose parameters. Variances are plotted against error in pixel coordinates, with a maximum variance of 5 pixels which corresponds

to approximately 1% of the image size. As can be seen, translation and normal estimations are adversely affected by image noise. The errors for the other algorithm [51], were similar.

The variances in Figures(3.1a) plot the error in estimation of rotation parameters when noise is introduced into the system. As is seen, the maximum variation of rotation parameters in the Euler angle space is 6 degrees, for as high as one percent image noise. Comparison with the translation and normal errors, which are as high as 40 degrees in the polar space Figures(3.1b-3.1c), show that the decomposition algorithm produces much more robust estimates of rotation than either translation or normal parameters. This explains the greater need for better estimates of translation and normal parameters compared to that of rotation parameters that are much close to the actual values.

3.3.2 Implementation Issues and Sensitivity Analysis

The implementation of both decomposition algorithms start with the SVD of H and thereafter, a sequence of `if-else` conditions on the resulting eigenvalues gives rise to various ways of computing the different parameters $\{\mathrm{R}, \mathrm{t}, n\}$ from these values. The only point to note is that in the implementation of the algorithm of Faugeras [51], a scaled Homography matrix is passed along with a point m_1 on the plane such that $m_2 = \mathrm{H}m_1$ is an equality and not an equivalence relationship ($\lambda = 1$). Ofcourse, both all the quanitites passed as input to both these algorithms are first normalized with respect to the internal parameters of the camera.

Sensitivity Analysis

Error in Homographies that are decomposed to obtain pose, may be from two sources. The first one is the well known error in image correspondences, and the second is the error introduced due to manual or auto-calibration of the views involved. If the standard RANSAC approach [1] is used to compute Homographies, then the error in Homographies as a result of error in image correspondences can be approximated by a Gaussian to the first order [110]. This is done by using a theorem established earlier, that measures the perturbation in the eigenvalues and eigenvectors of a matrix, as a function of the perturbation in the matrix elements themselves [111].

The reason why extending this approach to study error in pose from Homographies is infeasible, is because the computation of pose from image correspondences through the computation of Homography requires not one but two SVDs. Although in the case of Homography computation [110], the Homography is directly the eigenvector with least eigenvalue of the matrix in consideration, pose and normal values turn out to be non-linear functions of the eigenvalues of H. Secondly, the theorem in [111] only gives a first order approximation of the error, and so extending it for studying error in pose is of less practical use. Thirdly, this method for studying error is only correct when

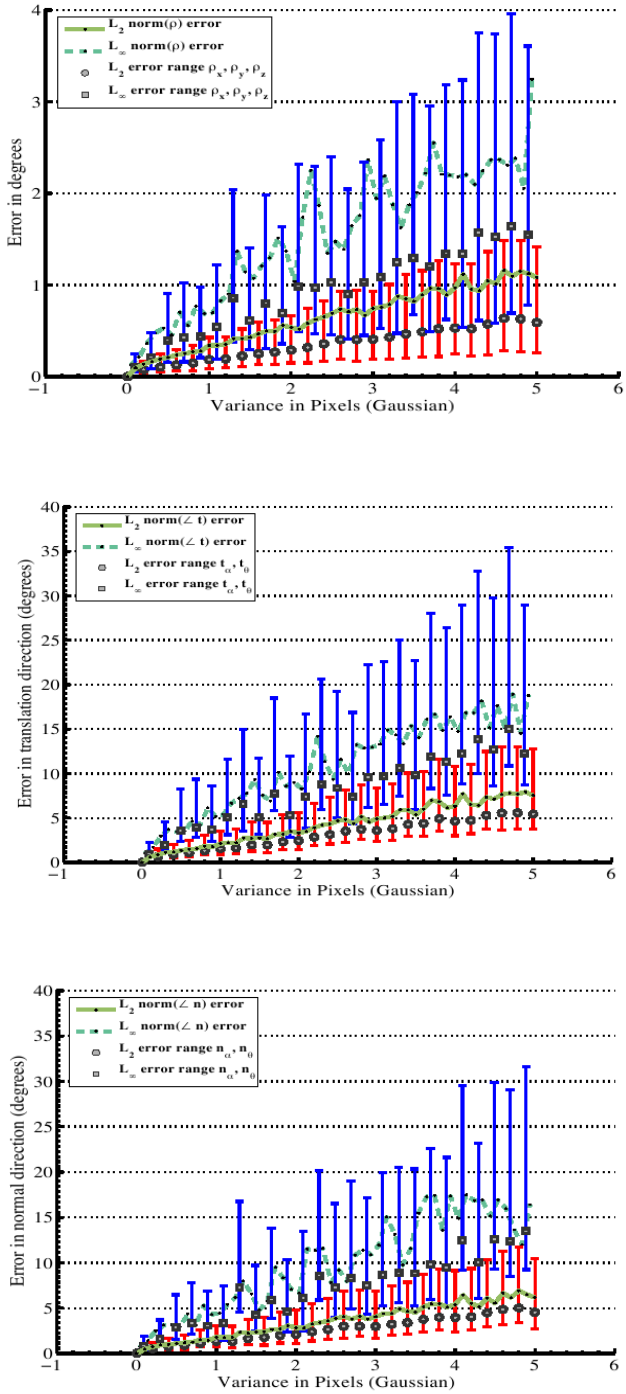


Figure 3.1: (a,b,c) Plot the $L2$ and L_∞ errors in the rotation angles, translation direction and normal direction respectively. Also are plotted the maximum error ranges for these quantities. The translation and normal direction errors are computed as Euclidean distances in polar space.

a RANSAC based approach is used for Homography estimation. Finally, calibration errors are not accounted for in this approach.

Thus it is more advantageous to do an empirical study of the error in the estimation of plane and pose parameters, given noise in image correspondences and calibration. As of now, we restrict ourselves to study errors arising from image correspondences alone. Extending it to calibration errors is a useful topic for future study.

3.4 Convex Framework for Planar Reconstruction

In this section, we formulate the problem of planar reconstruction using homographies in a convex optimization framework. We propose an algorithm for planar reconstruction in videos, the algorithm has no constraint that all the planes should be visible in all the frames. We also show how we utilized the inter-image homographies as additional constraints on the algorithm, which makes the method robust. We also discuss the issues with the current algorithm.

3.4.1 Formulation of the Objective Function

We wish to find plane and pose parameters that minimize a suitable variation of the difference between the L.H.S and R.H.S of Equation 3.1. Observe that the relationship in Equation 3.1 is non-linear in terms of the quantities $(\mathbf{R}, \mathbf{t}, n^j, d^j)$, which are the parameters we need to compute. However, if either the camera pose or the plane parameters are known, the above equation is linear in terms of the rest of the unknowns. Thus we define the following objective functions(in equations (3.3, 3.5)) that measures the geometric distance between the computed plane-pose parameters and the homography estimated from point correspondences, for the jth plane.

$$\mathcal{H}rt^j = \left[\mathbf{R} - \frac{\mathbf{t}n_c^j}{d_c^j}\right] \tag{3.2}$$

$$\mathcal{F}_{(\mathbf{R},\mathbf{t})} = \sum_{i=1}^{8}\left(\frac{H_i^j}{H_9^j} - \frac{\mathcal{H}rt_i^j}{\mathcal{H}rt_9^j}\right) \tag{3.3}$$

$$\mathcal{H}nd^j = \left[\mathbf{d^j}R_c - t_c\mathbf{n^j}\right]) \tag{3.4}$$

$$\mathcal{F}_{(\mathbf{n},\mathbf{d})} = \sum_{i=1}^{8}\left(\frac{H_i^j}{H_9^j} - \frac{\mathcal{H}nd_i^j}{\mathcal{H}nd_9^j}\right) \tag{3.5}$$

Here (R_c, t_c, n_c^j, d_c^j) denote constants and letters in bold denote variables whose values need to be computed, and elements of all homographies are accessed in column major order. There are

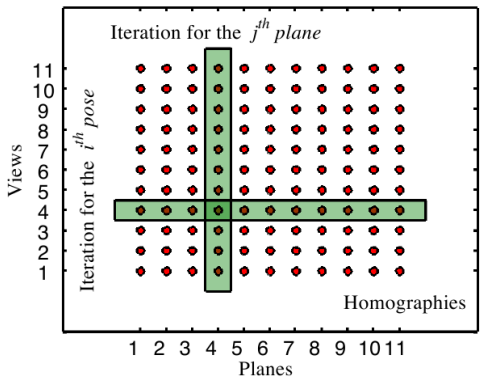


Figure 3.2: Proposed algorithm: Each dot in the above figure represents a homography H_i^j . An iteration for refining i the pose of a single view minimizes over data from all the planes, and an iteration for refining a single planes parameters minimizes over data from all the views.

two issues to be noted about equations (3.3, 3.5). Firstly, both these equations are linear fractional equations: both the numerator and denominator are affine functions of the unknown parameters. Secondly, it is possible to optimize all the parameters by iterating Equation 3.3 and Equation 3.5 alternatively till convergence. This is summarized Algorithm 1.

3.4.2 Proposed Algorithm

The proposed algorithm traces through two steps for the estimation of pose parameters given the Homography. The first step is to acquire an initial estimate using an SVD-based decomposition. Then scale issues related to the decomposition are resolved Section 3.4.3. The values of (R^j, n^j, d^j) are used to initialize the search for a global estimate of t^i, which is then subsequently used to search for global estimates of (n^j, d^j).

The second step using convex optimization, is an iterative process that refines t^i in one step and (n^j, d^j) in the following step as show in the Figure(3.2). Since, each of the plane parameters are independent of the other, and the pose parameter for each view is independent of the other, optimizing all the variables together has the same effect as optimizing for each view and each plane separately. Thus, optimization of t^i takes into account information from the homographies induced by all the planes $\mathrm{H}_i^{1\ldots m}$, and similarly optimization of (n^j, d^j) takes as input all the homographies $\mathrm{H}_{1\ldots k}^j$. This is done in a two step process to ensure the quasiconvexity of the two minimization problems.

Algorithm 1 Complete Algorithm Summarized.

1: Input: Homographies ${}^kH_j$ for $j = 1, \ldots, J$ and $k = 1, \ldots, K$ of plane Π_j between the camera views ${}^kP$ and reference view ${}^0P = I$.

2: SVD-based decomposition: Decompose ${}^kH_j$ to get ${}^kR_j$, $\frac{{}^kt_j}{{}^kd_j}$, ${}^kn_j$.

3: Initialization: ${}^kR =$ median$_j$ $\{{}^kR_j\}$ and $t =$ median$_j\{{}^kt_j\}$.

4: Set to universal scale: Assume each actual camera translation to be a unit vector in the direction of $\frac{{}^kt}{d_j}$, i.e., $\|{}^kt\| = 1$. Let ${}^kG_j = [{}^kR + \frac{{}^kt{n_j}^T}{{}^kd_j}]$ and ${}^kG_j^s = (g_1, g_2, \ldots, g_9)^T$.

5: Iterative Minimization:

6: $\Sigma_k\Sigma_j \left\{ {}^kH_j^s - {}^kG_j^s \right\} \leq \delta$

7: Update (R, t): $(R, t) = \arg\min_{{}^kR, {}^kt} \Sigma_j\Sigma_i[\frac{{}^kh_i}{{}^kh_9} - \frac{{}^kg_i}{{}^kg_9}]^2 \forall k = 1, \ldots, K$.

8: Update (n_j, d_j): $(n_j, d_j) = \arg\min_{n_j, d_j} \Sigma_k\Sigma_i[\frac{{}^kh_i}{{}^kh_9} - \frac{{}^kg_i}{{}^kg_9}]^2 \forall j = 1, \ldots, J$.

3.4.3 Discussions

Proof of Convergence

We show that the value of the objective function either decreases or remains constant at each iteration. The function being minimized is $|\mathcal{F}_{(x)}|_\infty$, x being the variables over which optimization is performed. The iterative minimization process (step 5 in Algorithm 1) is a two step process. In the first step minimization is over (R, t) and the second step is over (n, d). Given an initialization (R_i, t_i, n^j, d^j) if we prove that in each iteration the value of the objective function does not change it would be sufficient to explain that the algorithm converges to the (local) minima. We observe the following two corollaries.

Corollary Given an initial point $x_I = (R_i, t_i, n_j, d_j)$ the value of the functions $\mathcal{F}_{(t)}$ and $\mathcal{F}_{(n,d)}$ either decreases or remains constant for each iteration in the minimization i.e., $\mathcal{F}(x*) \leq \mathcal{F}(x_I)$.

Note that the objective functions in steps 7 & 8 are the same except for the scale factor that has no effect on the minimization process. Thus the proof is easily seen, since the L_∞ based quasi-convex function is minimized to find a global minima in each step. Since the output of one step is given as input to the next iteratively, we see that with every iteration the geometric error either increases or remains constant. However, with the non-linearities associated with rotation parameters and the fixed point iterative solution suggested, existence of a global optima is not direct.

Universal Scale

Each decomposition by the algorithms of Faugeras and Zhang produces estimates of $\{R, t, n\}$ assuming a coordinate system in which the perpendicular distance between the origin and the plane in consideration is 1. Since we consider all the homographies computed with respect to a fixed reference frame, the origin in all the decompositions obtained is the same. Thus the difference in the various solutions obtained by SVD decomposition differ in a scale factor, which in the presence of noise has to be computed using optimization.

Let the solutions of translation obtained by decomposition methods be denoted by t_i^j , which is the translation vector obtained by decomposing the homography H_i^j. Thus the actual translation vector is represented by $t_i = t_i^j d^j$, where d^j is the optimum of an objective function. Since, estimates obtained from the various planes must converge, we are interested in the optimum values $\left[d^{*1}, d^{*2}, \ldots, d^{*m}\right]$ such that

$$\left[d^{*1}, \ldots, d^{*m}\right] = \min \sum_{j=1}^{k} \sum_{l=1}^{k} \sum_{i=1}^{m} |t_i^j d^j - t_i^l d^l|_2 \tag{3.6}$$

The above function is quadratic and can be reduced to the form $|\mathbf{Ax}|$. However, we wish to not only find an approximate solution for the perpendicular distances, but also to get an estimate of the translation of the current frame, which can then be used for initializing the convex optimization routines. For this task we introduce a new set of variables $(t_i, i = 1, \ldots, k)$ which represent the *actual* translation of the i^{th} view upto scale. The modified functions now become $f_{i,j}(t_i, d^j) = |t_i - t_i^j d^j|_2$. As can be seen, these set of functions can be re-written in the form

$$f_{i,j}(t_i, d^j) \quad = |t_i - t_i^j d^j|_2 = \quad |A_{i,j} x_{i,j}|_2 \tag{3.7}$$

$$A_{i,j} \quad = \left[I_{3\times 3} \quad -t_i^j\right], x_{i,j} = \quad \begin{bmatrix} t_i \\ d^j \end{bmatrix} \tag{3.8}$$

Instead of minimizing the sum of square errors of all the functions $f_{i,j}$, a convex formulation may be obtained by minimizing the *maximum* of these functions. Since the functions $f_{i,j}$ can be thought off as the composition of a norm function and an affine function, its easy to show that these functions are convex in nature. Since convexity is preserved under point-wise supremum [80], we can collect the required variables and functions into one framework.

$$x = \left[t_1 \dots t_k \, d^1 \dots d^m\right]^\top \tag{3.9}$$

$$\begin{aligned} \left[t_1^* \dots t_k^* \, d^1 \dots d^m\right] = & \arg\min \quad \gamma \\ & \text{s.t} \quad \max_{i,j} f_{i,j}(A_{i,j}x) \leq \gamma \\ & i \in [1 \dots k], k \in [1 \dots m] \\ & A_{i,j} \in \mathbb{R}^{3\times 3k+m} \end{aligned} \tag{3.10}$$

In the above formulation, unconstrained optimization would produce the solution x with all zeros. Since this is undesirable, we *fix* one of the perpendicular distances (say d^1 without any loss of generality) to 1. This also sets the overall scale of the minimization process, and since functions $f_{i,1}$ are now reduced to the Euclidean norm function, it *moves* the optimization process away from the other pitfalls, towards the correct solution.

Algorithm 1 is a consequence of the structure of the relationship between a homography and the corresponding plane and pose parameters, and allows us to integrate information about planes across views into one minimization framework. A parallel can thus be drawn between the current framework for planes and the traditional bundle adjustment algorithm, for points. However, for this analogy to be complete, two important issues remain to be considered. First is the estimation of rotation, which we have sidelined until now. The second is the inclusion of planes *not observed* in the first image. These related issues are discussed in the next section.

3.4.4 Additional Constraints

We extend the framework described previously to include two important aspects: the estimation of rotation and the inclusion of inter-image homographies as additional constraints. An additional advantage of adding inter-image homographies is the tightening of bounds of the optimization process.

In effect, we intend a graph based estimation of homography like ones presented in the mosaicing literature [112] to be a precursor to our algorithm. Thus, outlier homographies can be identified and thrown away by graph based approaches, and the remaining homographies can be used to find optimal solutions to the pose and plane parameters

Consider a homography $\mathrm{H}_{i,k}^j$ that is induced by the j^{th} plane between the i^{th} and the k^{th} cameras. It can be broken into the following equation

$$\mathrm{H}_{i,k}^j = \mathrm{R}_i^k - \frac{\mathrm{t}_i^k n_i^{j\top}}{d_i^j} \tag{3.11}$$

where the subscript i denotes that all quantities are measured keeping the i^{th} frame as reference (origin). These quantities are related to the actual reference coordinate as

$$\begin{aligned} \mathrm{R}_i^k &= \mathrm{R}_k \mathrm{R}_i^\top & (3.12) \\ \mathrm{t}_i^k &= -\mathrm{R}_k \mathrm{R}_i^\top \mathrm{t}_i + \mathrm{t}_k & (3.13) \\ \mathrm{n}_i^j &= \mathrm{R}_i \mathrm{n}^j & (3.14) \end{aligned}$$

Given that $\mathrm{H}_{i,k}^j$ can be computed and hence decomposed accurately, the above equations provide additional constraints on $\{\mathrm{t}^i, \mathrm{t}^k, \mathrm{n}^j\}$ which can be formulated as the minimization of the square difference between left hand and right hand side quantities. The most important result of adding such additional constraints is that it allows us to include additional planes in the optimization process that are *not visible* in the reference frame. As will be seen later, these constraints also provide, much needed robustness to outliers, since the L_∞ norm is known to be susceptible to them.With these additional constraints, we now have an algorithm that optimizes *all* the tracked planes and views of a video sequence, robustly.

3.4.5 Issues with Rotation and Normal

The primary issue with rotation and normal parameters in the objective function are the constraints associated with them. The norm constraints on the rows and columns of the rotation matrix, as well as on the normal are not convex. Thus, at present, our algorithm solves a relaxed version of the original problem for normals. In literature [101, 113], this issue has been solved by modifying the problem with constraints that are under estimators and over estimators of the actual non-convex function, in a Branch and Bound algorithm.

In order to extend this approach to the problem of plane based pose estimation, we need to introduce the image coordinates of the planes concerned, into the objective function constraints. To do this, let us observe that an alternative to the currently used objective function Equation 3.3 is to consider minimizing the angular distances between image points transferred using the measured homography, and those transferred due to the homography computed from pose estimates. More precisely, let us consider the objective function

$$\mathcal{F}_{(\mathbf{R}_i, \mathbf{t}_i)} \equiv \mathbf{Find}(R_i, \mathbf{t}_i) \qquad \text{s.t. } \angle(H_i^j \mathbf{x}_1^j, (\mathbf{R}_i - \mathbf{t}_i \frac{n^j}{d^j}) \mathbf{x}_1^j) < \epsilon_{min} \tag{3.15}$$

which can be alternatively posed as

$$\mathcal{F}_{(\mathbf{R}_i, \mathbf{t}_i)} \equiv \mathbf{Find}(\mathbf{R}_i, \mathbf{t}_i) \qquad \text{s.t. } \angle(H_i^j \mathbf{x}_1^j, \mathbf{R}_i(\mathbf{I} - \mathbf{t}_i \frac{n^j}{d^j}) \mathbf{x}_1^j) < \epsilon_{min} \tag{3.16}$$

In the objective function proposed above, $\mathbf{x}_1^j$ represents the points belonging to the jth plane in the first view. The transfer of points from the first view is chosen over the points detected in the ith view primarily to eliminate errors due to the feature detection and tracking process, which can be considered even at the homography estimation stage [?]. Arguments of bounds and in general the branching strategy of [101] can now be incorporated into the current framework.

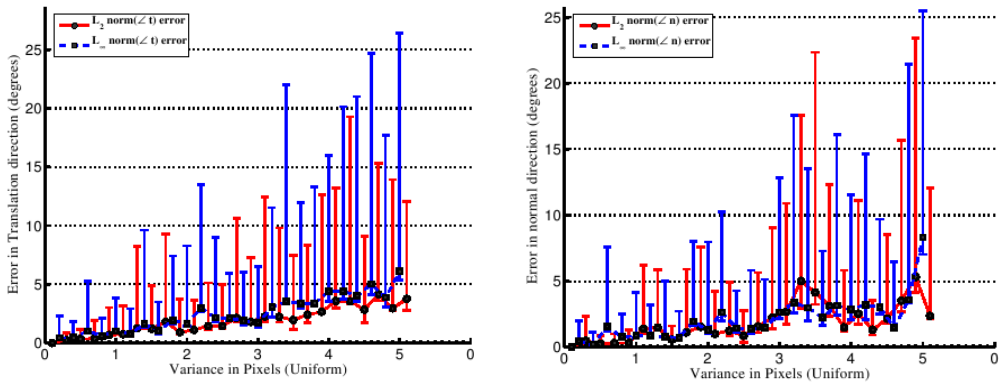
3.5 Experimental Analysis

In order to test the proposed algorithm, we have designed experiments on both synthetic and real-world data. Synthetic data is obtained by generating points on planes and projecting them onto camera matrices. Real world data sets tested include the Oxford Model House, Corridor, and UNC datasets. In all these cases, the real world is assumed to be segmented into planes apriori *i.e,* interest points and hence correspondences computed are assumed to be clustered into planes. However, there are automatic algorithms to achieve such a classification [104] .

3.5.1 Synthetic Data

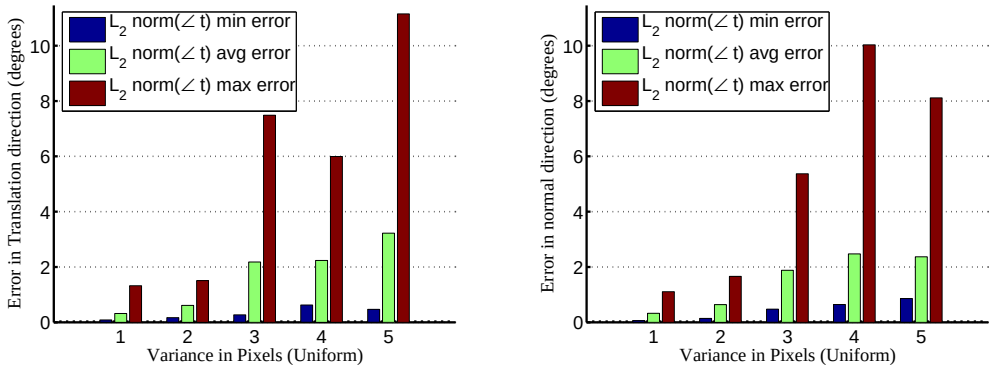
Generation Random points are generated on the XY-plane which is then re-positioned at a random location. Two random camera matrices are generated and the world points of many such planes are projected using them to generate image points. Gaussian noise of varying standard deviation is added to these image points to create synthetic correspondence data. Homographies are then computed using the RANSAC after normalization [1] which can alternatively be generated by [81]. The generated Homographies are decomposed using Faugeras' and Zhang's algorithms [51, 52] to generate data for both initialization and comparison. Algorithm 1 is then run with this data, to produce our estimate and is compared with the SVD algorithms and Bundle Adjustment in the 6-parameter pose space by plotting the euclidean distance between estimated and ground truth values.

Experiment 1: Effect of noise Figures (3.3a,3.3b) show the effect of increasing image noise on the accuracy of estimation. Two effects can be observed for both translation and normals. First, the average error in the estimation of both parameters is less than 5 degrees even for a 1% error in the image coordinates, which is a serious error. This justifies the robustness of our algorithm to image noise. The second effect is that the mean errors (averaged for 100 trials) in all these cases are located close to the minimum errors represented by the lower end of the error bar. Figures (3.4a,3.4b) show that most of the estimations center around the mean, with only a few deviating towards the higher end. Another interesting observation is that even the resilience to noise is apparent till about 3 pixel error after which the maximum error in both cases seems to increase. This can be attributed to the



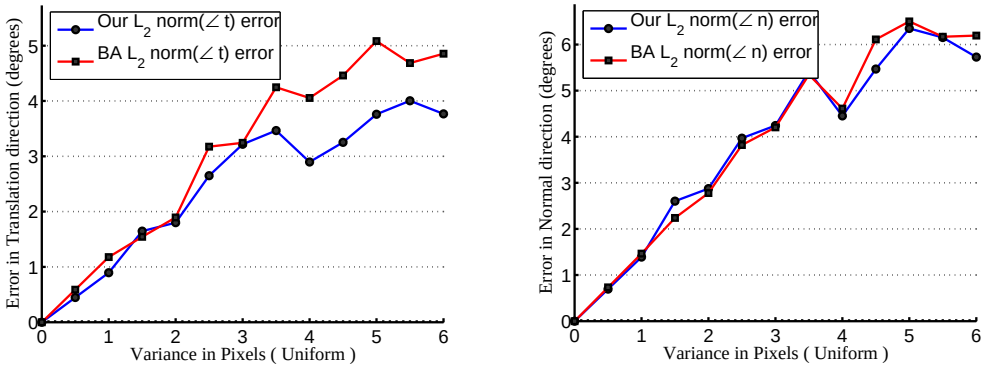
(a) Translation (b) Normal

Figure 3.3: Plot of L_2 and L_∞ norms of the distance in pose space between estimated and ground truth quantities from Algorithm 1 against increase in variance of Gaussian error in point correspondences. Comparison with two SVD based methods is shown.



(a) Translation (b) Normal

Figure 3.4: Plot of minimum, average and maximum of L_2 norms of the distance in pose space between estimated and ground truth quantities from Algorithm 1 from 100 trails against increase in variance of Gaussian error in point correspondences.



(a) Translation (b) Normal

Figure 3.5: Plot of L_2 norm of the distance in pose space between estimated and ground truth quantities from Algorithm 1 and Bundle adjustment against increase in variance of Gaussian error in point correspondences.

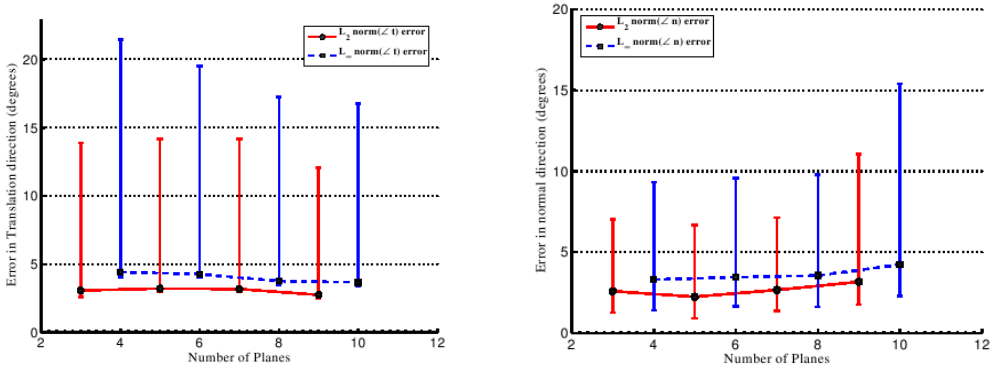
fact that after a point the algorithm possibly settles into a local minima because of the inaccurate initialization. However, this is still far better than the SVD decomposition in Figures 3.1b, 3.1c.

Experiment 2: Comparison with Bundle Adjustment We empirically compare our algorithm with standard iterative non-linear optimization technique of Bundle Adjustment [59](See section 2.1.5), which uses Levenberg-Marquardt internally. Bundle adjustment is initialized by the output of the SVD-based approaches similar to our case.

The improvement in translations is shown in Fig (3.5a) and that of normals in Fig (3.5b). They are shown for varying levels of variance each of which has been tested for 100 trials. This clearly shows that our algorithm is better than Bundle Adjustment.

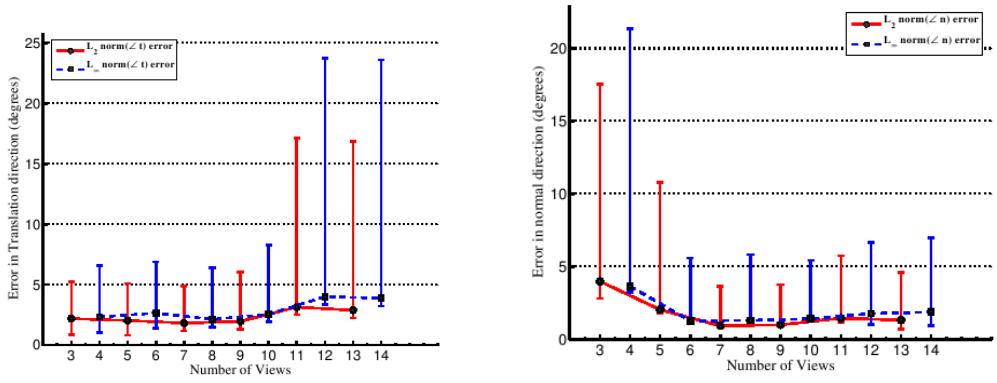
Experiment 3: Effect of planes Figures (3.6a,3.6b) show the effect of the increasing number of planes on the overall result. Contrary to expectation, increasing the number of planes does not seem to have much effect either on the accuracy in estimation of translation parameters, nor the estimation of normal parameters.

Experiment 4: Effect of views Figures (3.7a,3.7b) show the effect of increasing the number of views, in this experiment the number of parameters increases significantly and hence accuracy in the translation errors dwindles down . In the case of normals, as expected, increasing the number of views results in a marked improvement in the accuracy of the estimated normal values.



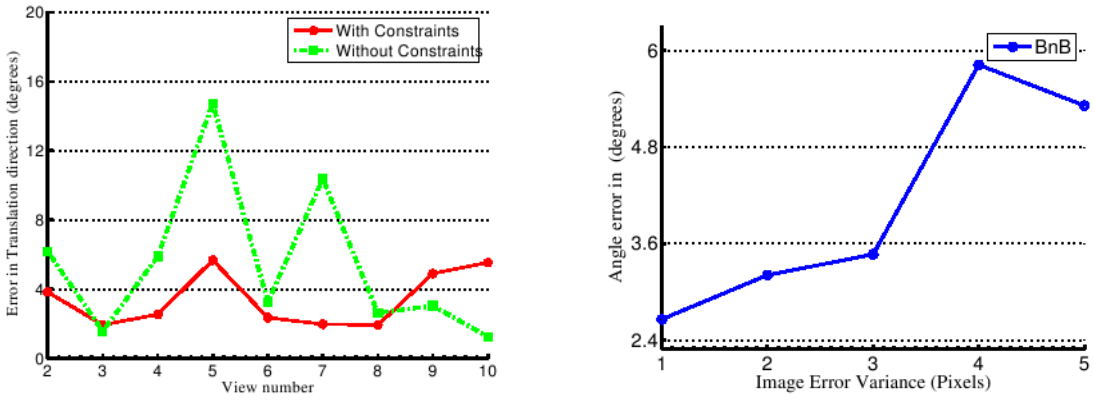
(a) (b)

Figure 3.6: The above figures plot the effect of planes on the accuracy in estimation of the translation and normal parameters respectively. In this experiment we varied the number of planes from 2 to 10 and the number of views was kept constant at 10



(a) (b)

Figure 3.7: The above figures plot the effect of views on the accuracy in estimation of the translation and normal parameters respectively. In this experiment we varied the number of views from 3 to 15 and the number of planes was set to be 3.



(a) (b)

Figure 3.8: (a) Shows the improvement of estimating translation parameters using additional constraints, when a single plane has bad homographies. (b) Shows the estimation accuracy of rotation parameters, using the branch and bound algorithm. The estimation is accurate and robust.

Experiment 5: Effect of Extensions Figures (3.8a,3.8b) show the effect of adding inter-image homographies as constraints (Figure 3.8a), and the accuracy of the branch and bound algorithm for estimation rotation (Figure 3.8b). As expected, inter-image homographies produce tighter bounds around the global minima of the pose parameters, preventing the optimization algorithm from fitting outlier data (Figure 3.9). This results in better accuracy in estimation and resilience to noise than the unconstrained case. The computation of rotation parameters using the modified branch and bound algorithm [101] produces accurate estimates, with good robustness to noise.

3.5.2 Real Data

In order to test on data from the real-world, we chose datasets, of which two are Oxford data sets and the other one is UNC dataset. The House, and Corridor data sets (Figures (3.10a,3.11a)) are accompanied by correspondences and estimates of the camera matrices produced by other robust estimation algorithms and hence provide a good benchmark with which to compare our algorithm's performance.

Figures 3.10b-3.10c show the comparison between our estimation and that of the decomposition of Faugeras. The L_2 and L_∞ errors between the estimated and ground truth quantities are plotted. In order to compare the plane normals, we took the best estimate of normals from the several decompositions available. As can be seen from the results, our algorithm produces far better estimates for the translation parameters than the corresponding algorithm by Faugeras. We found that Zhangs

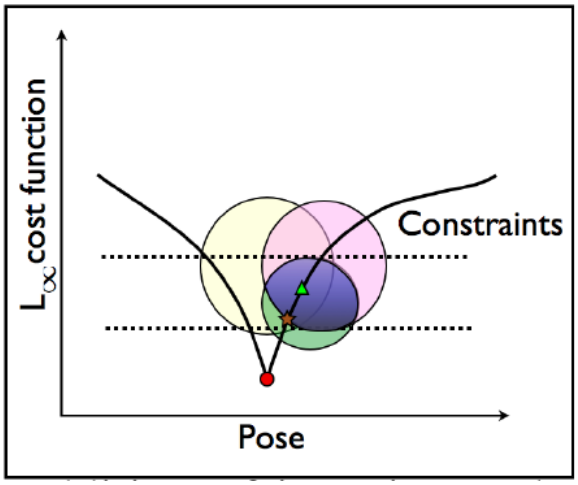
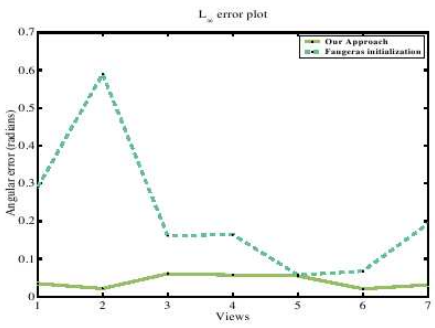
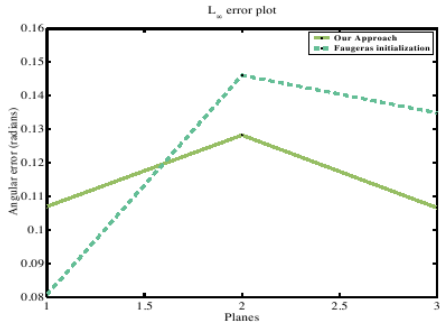


Figure 3.9: The addition of inter-image homography based constraints improves the robustness of the system. The current cost function is designed to overfit outliers. In the above figure, while the red circle represents the minima corresponding to the error function, the actual global minima, the green triangle represents the global minima while the brown star represents the solution with constraints. Each of the circles represents constraints, and the accuracy of the resultant solution depends on their tightness.

(a)



(b)



(c)

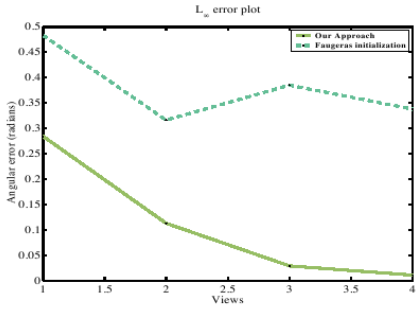
Figure 3.10: ***Dataset 1:*** Oxford-house dataset (a) Sample image from the dataset. (b-c) Plots of the L_∞ error between plane and pose parameters with respect to the ground truth L_2 error shows similar plots.

algorithm also produces similar estimates to Faugeras in most cases. The same situation is repeated in the Corridor sequence (Figures 3.11b-3.11c), where translation is very accurately obtained. An explanation of why certain plane parameters are perturbed to a value of higher error is that since some of the homographies are erroneous, the error in a particularly bad homography is distributed across planes.

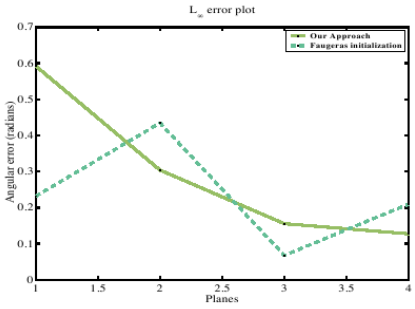
3.6 Discussion

In this chapter, we have proposed a framework that produces reconstruction of piecewise planar scenes in much the same way as Bundle Adjustment for point sets. The algorithm incorporates both multiple planes and views, and does not constrain all the planes to be visible in any single view.

(a)

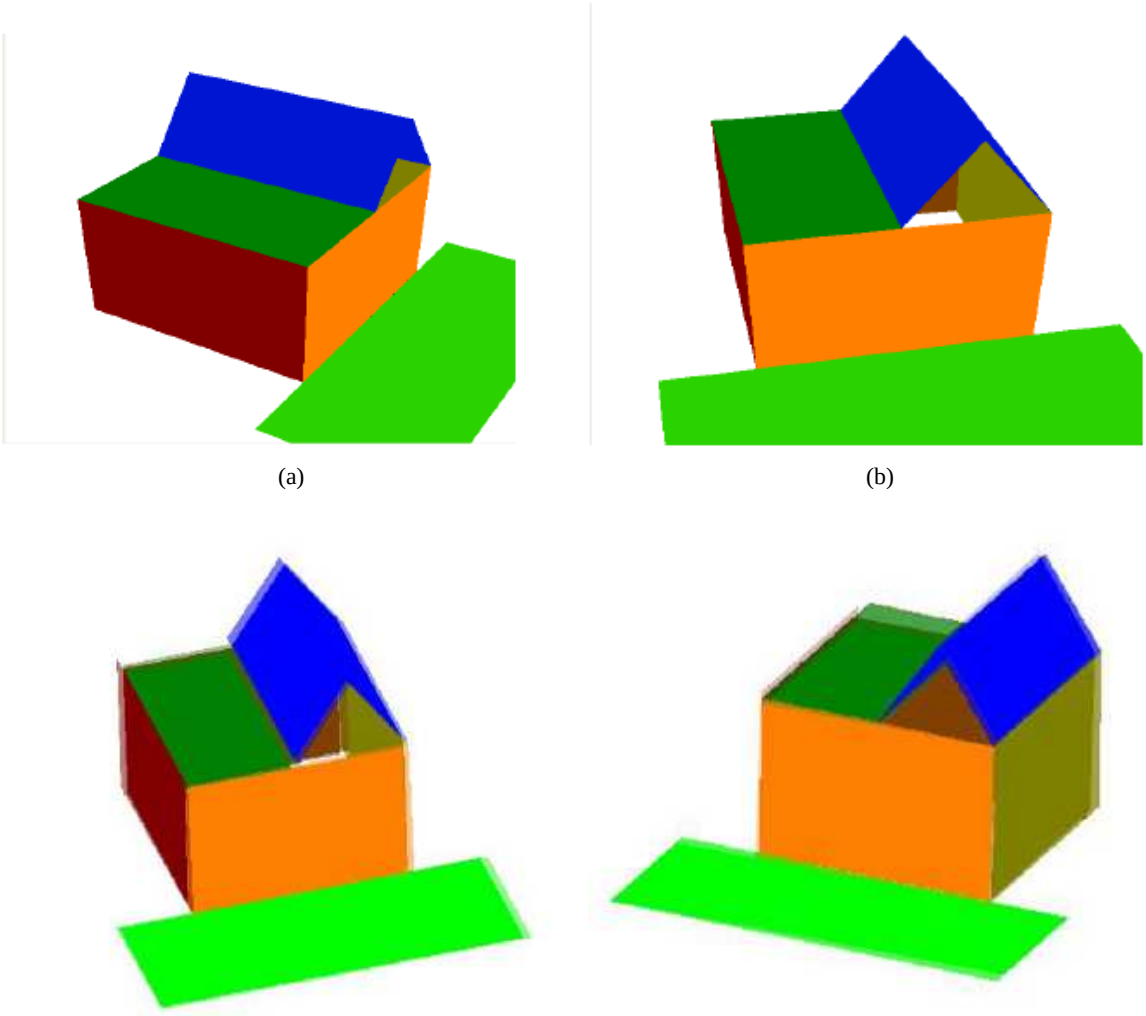


(b)



(c)

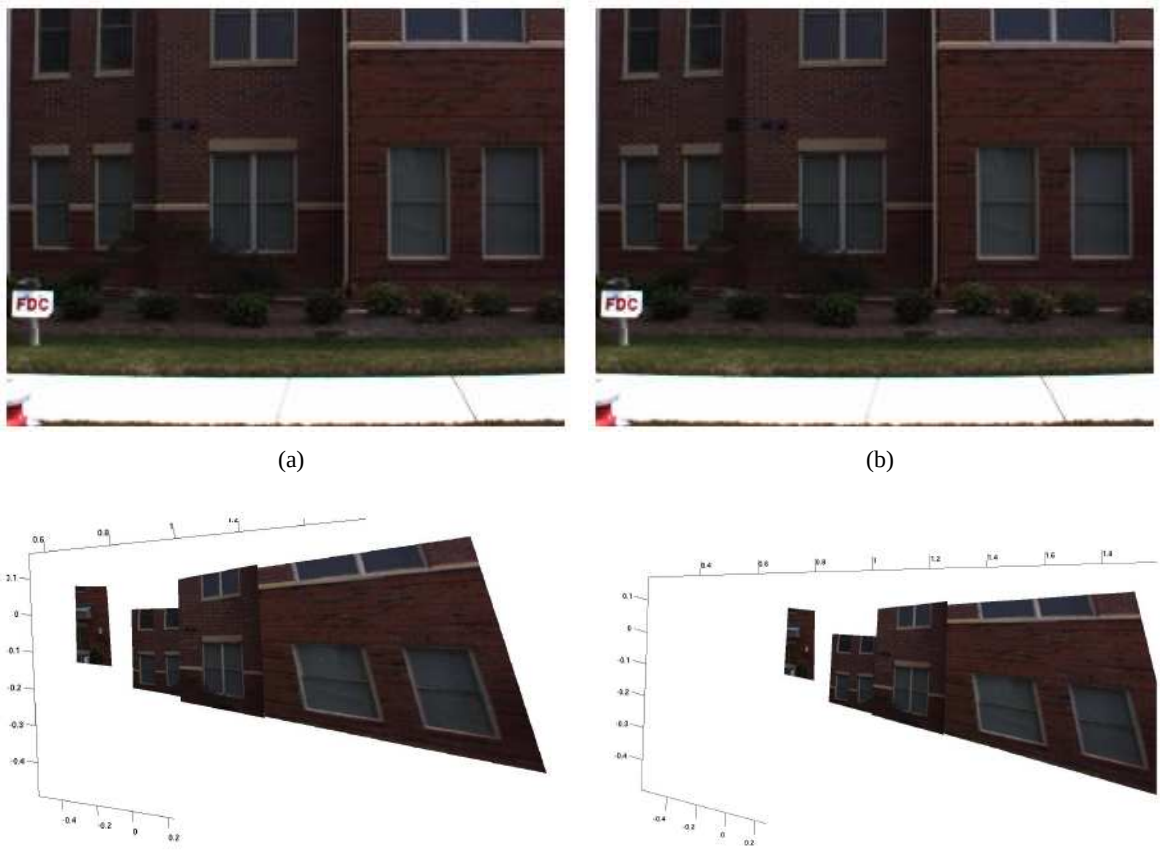
Figure 3.11: ***Dataset 2:*** Oxford-corridor dataset (a) Sample image from the dataset . (b-c) Plots of the L_∞ error between plane and pose parameters with respect to the ground truth L_2 error shows similar plots.



(a) (b)

(c) (d)

Figure 3.12: ***Dataset 3:*** Synthetic house (a-b) Sample images from the dataset . (c-d) Illustrates the accuracy of our reconstruction. The ground truth and reconstructed models are overlapping to a greater extent



(a) (b)

(c) (d)

Figure 3.13: ***Dataset 4:*** UNC dataset (a-b) Sample images from the dataset . (c-d) Texture mapped reconstructions of UNC dataset.

Additionally, the presence of inter-image homographies presents useful robustness to outliers, that may not have been pruned in the initial stages of registration and homography computation.

The existing framework is not without its drawbacks. Currently, though the objective functions show robustness to noise, it has not been systematically incorporated into the objective functions. Existing literature on robust convex optimization may be used for this purpose [114]. Secondly, constraints *between* planes may help in stabilizing the overall reconstruction [108], like orthogonality of planes. One other issue related to this algorithm is its practical applicability. Recent results in Practical Global Optimization [105, 115] is very relevant to our work, and may be used to improve the running time of our algorithm, making it suitable for faster computation required by videos.

Chapter 4

Terrain recognition using monocular camera

4.1 Introduction

In this chapter, our main focus of the study is to perform and analyze various experimental procedures that suits the problem of terrain extraction and recognition using only monocular camera. At the top level, our method consists of training phase and a testing phase, here we study various parameters that well suits these phases. We experiment with the size of the patch, that is optimal in representing the feature space as well as fast enough to be computed. We study popular feature extraction schemes, their richness in representing the feature in minimum possible size. We study various aspects of the spectrum of classifiers and their suitability with selected feature extraction schemes.

4.1.1 Contributions

In this chapter, we make the following contributions.

- We present our own annotated dataset, which contains huge varieties of scenes with various changes in environmental conditions. This data set allows us to conduct various experiments on our methods and it also allows us to compare with the state-of-the-art methods.
- We present extensive empirical comparisons of various features and state-of-the-art classifiers in machine learning literature.
- We also show how various parameters such as the richness of the features and the patch size that affect the classifier performance.

- This work along with the Partition-based algorithm in the next chapter were published in *International Conference on Pattern Recognition 2010* [116]

4.2 Problem Parameters

The problem of terrain characterization is that of essentially capturing the appearance of the surface from the images. This problem is modeled as a classification problem of pixels and smaller windows in the past [34, 90], where the important parameters of the problem are features and classifiers. We analyze the relative importance of these parameters on an annotated data set and demonstrate that the problem can be solved with state of the art features and classifiers. Though there are many new (and computationally expensive) features proposed in the recent past, we limit our attention to a set of simple and yet effective features due to their utility and aptness for the terrain characterization task.

4.2.1 Features

For any learning based method, selecting meaningful features for the classification task is very important. We use popular RGB histogram [4, 29] and LBP histogram [33] as our features considering the computational cost and performance. We use the optimal weighted combination of these features that best suits the classifier.

4.2.2 Classifiers

Performance of selected features are evaluated on a set of popular and promising classifiers. The baseline classifiers which we consider in our experiments are Naïve Bayes(NB), K-Nearest Neighbor(K-NN), Artificial Neural Networks(ANN), Support vector machines(SVMs) and Random Forests(RF) [99]. Random forest is a classification algorithm that uses an ensemble of unpruned decision trees, each of which is built on a bootstrap sample of the training data using a randomly selected subset of feature space dimensions. Experiments were conducted by changing important parameters like number of epochs and number of nodes in the hidden layers in ANNs, number of trees and size of node in RF. In case of SVMs, we conduct experiments with linear SVM using 1 vs 1 multiclass classifier (SVM-L) and non-linear SVM (SVM-K). From Table 4.1, we observe that RFs outperformed all other classifiers because of its capability to handle large number of input variables and data samples [99]. Additionally RF classifiers are computationally efficient for training and testing, compared to SVMs. Therefore we choose RF as our classifier.

Figure 4.1: Monocular camera attached at the top of the Van.

4.3 Data set and Experimental Setting

We argue that monocular camera-based terrain characterization solution have reached a state of acceptance in outdoor navigation. As a first step we do an empirical study on characterization performance and show comparable results on our dataset and other existing datasets. For consistency in evaluations, the performance of various features and classifiers, we build an annotated dataset.

4.3.1 Data set

Datasets are very important in determining the state-of-the-art of any research area. There are several datasets (for example [4, 117, 118]) introduced in literature in several fields of vision and robotics. However, as far as we know there is no dataset that is publicly available for the purpose of terrain classification. This motivates us to build our own dataset that is challenging and practical. Our dataset consists of road and off-road data, which may be used for terrain classification, scene segmentation, layer extraction, people detection and obstacle detection. For collecting data, monocular camera is mounted on the top of the vehicle (as shown in Figure 4.1), and videos were recorded by the camera at 7.5 fps and at resolution 800×600 on vehicle navigating at various speeds ranging from 0.2m/s to 4m/s. We set the camera to high aperture and high shutter speed, in order to minimize the artifacts caused by the moving camera like motion blur etc., We collect the data on ill-conditioned roads, in and around a radius of 10km, We observe that the data is challenging, as it contains wide variations in illumination. We also observe that the data varies from unpaved or

Figure 4.2: Overview of the dataset.

damaged rural roads to paved urban roads. Data also contains static (like trees, rocks) and dynamic obstacles (like moving vehicles). We collected 25 videos, each of 1 min. In total, we have collected 11250 frames.

Figure 4.2 shows some of the sample frames from the videos. We observe that the dataset contains huge variations in appearance. Five distinctly different terrains were identified in the data collection (see Figure 2.4):

- Road: This class consists of road patches which are mainly made up of tar or cement, we annotate these patches with black-grey color.
- Muddy-road: This class consists of patches of all kinds of mud. In constant white light, the color of the mud ranges from a tint of orange to brown. We annotate these patches with orange color.
- Rough-terrain: This class contain patches which are rough or rocky. Note that the mud in draught conditions falls into this class. We annotate these patches with brown color.
- Grass: This class contains only traversible grass or very small plants, big plants and trees are considered obstacles. We annotate these patches with green color.
- Obstacle: All the patches that doesn't belong to either of aforementioned four classes falls into this class. We annotate these patches with black color.

Figure 4.3: Patches from each of the identified classes.

Figure 4.3 shows random patches from the four identified classes. We observe that the variation in texture appearance in each class is quite high, simple color based classification is not sufficient. From all the recorded frames, 200 frames were randomly selected for experiment purposes. These images are hand labelled at pixel level using Interactlabeller [119]. After the labelling each image has its correponding annotated image as shown in the Figure 2.4

4.4 Classification procedure

As discussed in the previous sections, the fundamental task of terrain characterization can be formulated as an image classification and characterization problem. We start by exploring the performance of various features and classifiers discussed in Section 4.2. We consider a part of our data set (200 images) for the empirical studies. We use 50% of the data for training and the rest for testing. These images are manually densely annotated at pixel level as discussed in Section 4.3.1. From each of these annotated images, we extract multiple, non-overlapping, patches of size 16×16. Thus we have around 185000 patches for training, and a similar number of patches for testing. The number of patches in all the five classes is approximately equal for the initial studies. For all the randomly picked annotated training patches, we extract the features described in Section 4.2. We have experimented with various color and texture features mentioned in section 4.2. We have chosen one for each of the color, texture and combined features. We have chosen RGB histogram as color feature, because these are raw features and hence can be computed very fast. And we choose LBP histogram

as texture feature considering the computational cost and performance of various texture features. As a combined feature, we select optimally weighted color and texture features that best suits the given classifier. The combined feature always outperforms individual color or texture feature.

4.5 Experiments

In this section, we conduct several experiments to know the limitations and to get an overview of the performane of state-of-the-art machine learning methods for terrain classification. Specifically we show that monocular camera can provide useful characterization of the common terrains that can help in detection of navigable regions through features and classifiers delineated in section 4.2. We also experiment with few important parameters of the problem, that help in solving the problem to get best possible accuracies.

4.5.1 Experiment 1: Comparison accros classifiers

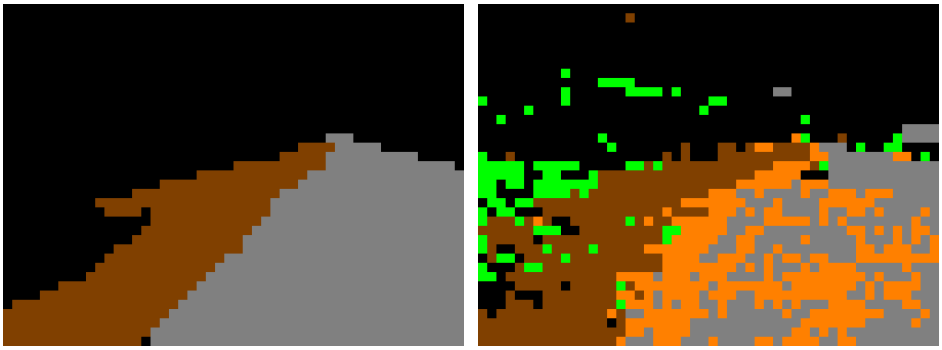
In this section, we compare the performance of different classifiers as well as features on our dataset and two other publicly available datasets [4]. The classifiers considered for the study are NB, ANN, K-NN, SVM-L, SVM-K and RF. Experimental results are shown in Table 4.1. It can be seen that RF classifier outperformed all other classifiers because of its capability of handling large number of input variables and data samples [99]. The other advantage of RF classifiers, is that they are computationally efficient for training and testing, compared to SVMs. SVM-K and K-NN performs moderately well, though training time for SVM-K is high, testing time is of practical importance, K-NN on the other hand has a very high classification time, unless approximate nearest neighbor computations are employed. NB performed the worst of all, it is due to its strong independence assumptions. In cases of certain features, the performance of SVM-K and RF are comparable. We also observe that though K-NN is computationally intensive, its performance is sometimes comparable to SVM-K.

Figures 4.4a, 4.4b and 4.4c shows the typical test image, the ground truth and the classification

Dataset	**NB**	**ANN**	**K-NN**	**SVM-L**	**SVM-K**	**RF**
Our	43.6	35.6	28.3	29.0	28.7	25.5
DS3A	18.9	32.3	33.8	31.2	38.4	18.2
DS3B	13.7	26.2	17.8	27.9	39.8	18.9

Table 4.1: Base line error-rates on Our dataset and two datasets of Procopio et al. [4].

(a)



(b) (c)

Figure 4.4: (a) Test image (b) Ground truth image (c) Labelled image using baseline RF classifier

result from the RF classifier respectively, we can observe that the base line result fails mainly due to illumination variations within the class, which some times makes little or no difference between patches from two different classes. This is mainly caused because, the spatial context is not being incorporated, we can also observe that the grass samples are being labelled at the upper portion of the image, which indicates that the baseline RF classifier cannot differentiate between the patches of the grass from that of the trees.

Since the data sets and details of the earlier reports are not completely available, a direct comparison of results may not be applicable. However, it may be noted that the quantitative results, which we report in Table 4.1, the performance of these methods on dataset due to [30] are comparable to to the results reported in literature [4, 30, 120], which use non-visual sensors and stereos along with appearance clues. We believe that this advantage comes out of the fact that monocular cameras in use now provide much richer sampling in space and dynamic range, and therefore useful for such tasks. This is specially true in contrast to the achievable resolution for laser and stereo. We also be-

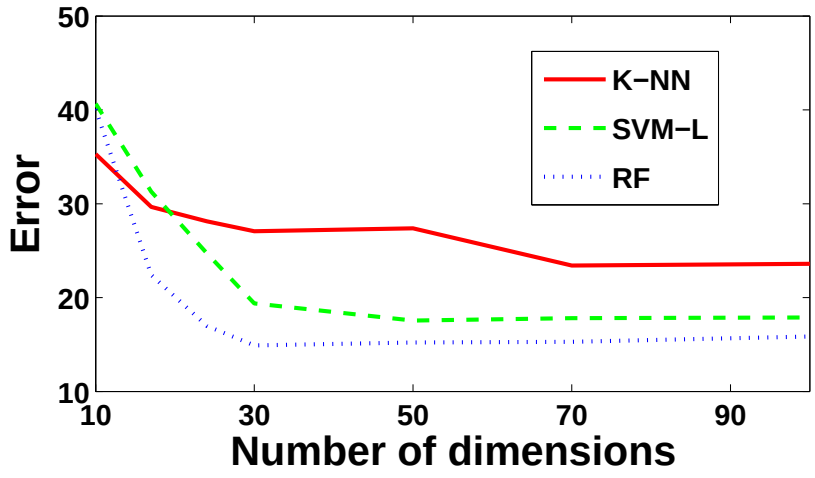


Figure 4.5: Experiment with increasing dimensions of combined feature on various classifiers.

lieve that our *reasonable* results are due to the use of diverse features possible form single frames of monocular images. We use a diverse and powerful set of features compared to many of the previous methods. We also show that with increase in features (diversity as well as dimensionality), we can obtain better classification.

4.5.2 Experiment 2: Effect on number of dimensions

The computational cost of training and testing a classifier is quite dependent on the dimension of the feature space used. Hence it is important to study the relationship between the dimension of the feature space and the performance of the classifier to choose the optimal feature space. In our training or testing phase, from each sample, i.e., from a patch, we extract features that are histogram of color and texture, histogram can be represented as a feature vector by quantizing it into fixed number of bins, the size of the bin depicts the richness of the feature vector. An experiment is conducted with varying the size of the bin. The relationship is shown in Figure 4.5. We choose K-NN, SVM-L and RF classifiers for this purpose, We observe that as the dimension of the feature vector increases, the error rate decreases and stabilizes at different error rates for different classifiers.

4.5.3 Experiment 3: Effect on patch size

As mentioned earlier, we use color and texture features for the terrain characterization. These features are evaluated at a coarse level (like from a patch), while the classification results are required at a finer (pixel) level for reliable navigation. Thus we explored the relationship between the windows at which features are extracted and the performance metric. The experiment was conducted

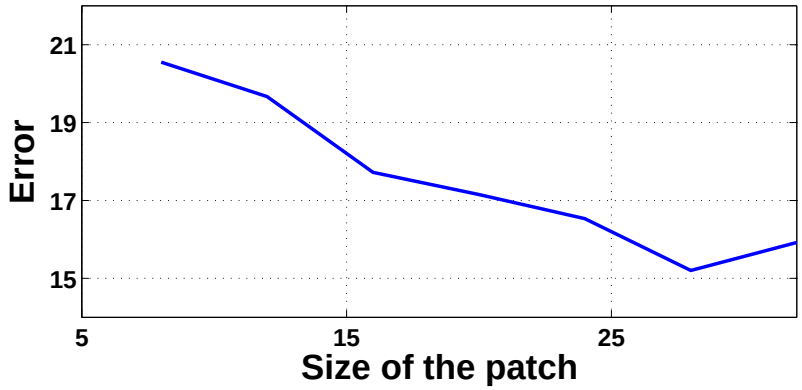


Figure 4.6: Error variation with varying window size.

for images of various sizes. The relationship between them is shown in Figure 4.6. We observe that the optimal size of the patch for the image of size 800×600 is around 28, we also observe that we can get improvement in the error as high as 5% by selecting the optimal sized patch. In general, we observe that the optimal patch size, that gives reasonable performance is approximately $(1/25)^{th}$ the size of the image.

4.6 Discussion

This chapter presented an annotated dataset in outdoor rural and urban terrains, which contain huge varieties of scenes with various changes in environmental conditions. This chapter reports extensive comparison of various classifiers operating on features for classification of outdoor terrains using only monocular camera. This chapter shows how various parameters such as the richness of the features and the patch size affect the classifier performance. The chapter reports that Random forests trained on weighted color cum texture feature gives the best baseline result, with an error of 25.5% compared to other classifiers such as Naive Bayes, Artificial neural networks, K-nearest neighbours and Support vector machines. On other publicly available dataset the baseline error rate was 18.2%. This chapter conducted various empirical studies with state-of-the-art machine learning techniques and various parameters of the problem.

Chapter 5

Fast and Adaptive Terrain recognition

5.1 Introduction

In the previous chapter, we analyzed various experimental procedures for terrain classification problem and we observed that the current state-of-the-art machine learning techniques achieve reasonable solution. We have observed that Random Forest classifier is performing best among several baseline classifiers. In this chapter we describe various enhancements for terrain classification. Initially we describe our partition based algorithm and several experiments which indicate that, the algorithm is robust and spatially smooth. Secondly we describe our label transfer method along with experiments showing that, it saves considerable amount of computation time. Subsequently we present our adaptive algorithm, which is designed specifically for videos and experiments show that it can adapt to slow appearance changes.

5.1.1 Contributions

In this chapter, we make the following contributions.

- We introduce our novel partition-based algorithm, which is build on random forest. We also conduct several experiments for the usability of the algorithm.
- We also introduce an adaptive-method which uses temporal information effectively using fast optical flow. The adaptive method is an online algorithm, which can adapt to fairly unseen terrains.
- This work was published in *International conference on Intelligent Robots and Systems 2010* [121]

5.2 Partition based algorithm

The proposed algorithm partitions the training images and trains different classifiers on different parts of the image independently. This is repeated for partitions of different sizes. Figure 5.1 pictorially shows the partitions in the image with respect to different sizes.Training different classifier from different part of the image handles the problem of perspectivity of the imaging process, i.e., it learns the fact that near and far image patches show different textural characteristics. Also learning from fixed partition over several training images has two main advantages. The first advantage is that it helps the classifier to learn new facts about associativity of classes, such as occurrences of grass along with mud is more probable than that of grass along with tar road. The second advantage is that it helps the algorithm to be dependent upon the position of the partition of the image and thus learns the spatial context. By training a classifier from larger sized partitions, global properties of the class are learnt and as the size of the partition decreases, more local properties are learnt. Our algorithm is a generic framework that can be operated on any classifier.

In training phase, as summarized in Algorithm 2 we build N classifier-sets, as the partition size increases from 1 to N, we have $\{ 1^2, 2^2, 3^2, ... N^2\}$ classifiers in each set respectively. Let us call them $S = \{C_1, C_2, C_3, ...C_{\mathrm{N}}\}$. Note that a classifier-set C_i contains i^2 classifiers. To characterize the terrain of the given image, for each patch of the image, we get N labels from each of the N classifier-sets in S. From these N labels, most occurring label is declared as the final label of the patch.

Implementation details: As mentioned before, we have 100 training and 100 testing images. For training Partition based algorithm, we need to build N classifier sets, each classifier set may be trained on all the patches from 100 training images. But this may create a problem of overfitting and also it increases the training time, to overcome this problem we randomly pick patches from the training set which are spatially distributed i.e., For each training image from the help of its ground truth image, we calculate the ratio of number of patches that belong to each class. Then based on those ratios we randomly pick patches from all training images such that there are approximately equal number of patches in each class. In our experiments, for each class we approximately have 1000 patches for training.

5.3 Experiments

In this section, we conduct several experiments to determine the capabilities and limitations of the proposed partition-based algorithm. Specifically we show that partition-based algorithm is a generic algorithm, that enhances the accuracy of any classifier. Secondly, we experiment with important

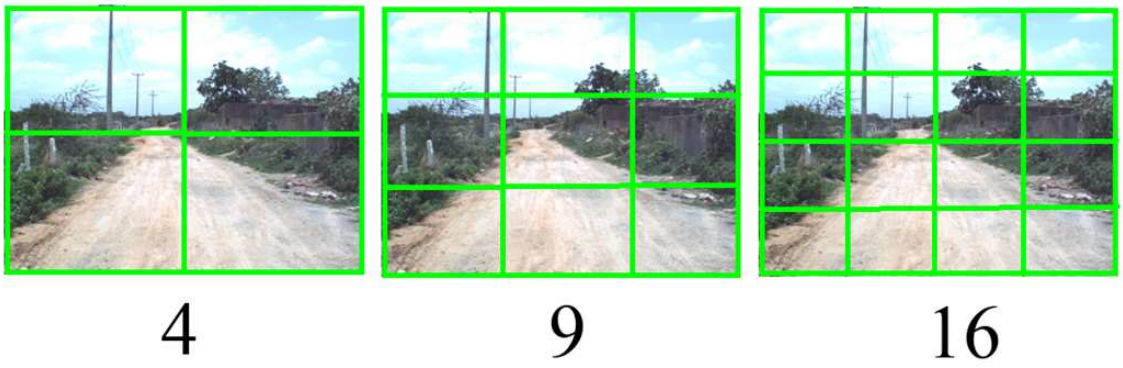


Figure 5.1: Pictorial representation of partitioning the images into 4,9 and 16 partitions respectively.

parameter of the algorithm *the number of Classifier-sets* and finally we perform another experiment which shows that the algorithm is capable of classifying the terrain spatially smooth avoiding costly post processing methods.

5.3.1 Experiment 1: Comparison with baseline classifiers

Figure 5.2a shows the percentage errors of our partition-based algorithm operating on baseline classifiers SVM and Random Forests. We observe that our algorithm always decreases the percentage errors by approximately 10%. This is an appreciable decrease in the percentage error. It also shows that our algorithm is generic, i.e., the algorithm improves the performance of classifier irrespective of the classifier chosen. To show the superiority of our algorithm across other databases, we conduct an experiment in which our partition-based algorithm operating over RF is tested on (i) Our dataset (ii) DS3A and (iii) DS3B datasets of Procopio *et al.* [4]. We report the percentage errors in first and second column of Table 5.1, from the table, we observe that our algorithm compared to baseline RF classifier, decreases the percentage error by approximately 10% on all three datasets. We also observe that even without training on any of the images of DS3A or DS3B datasets, we get percentage error as low as 6.8%, the superiority of our algorithm is thus clearly evident.

5.3.2 Experiment 2: Effect on number of Classifier-sets (N)

Figure 5.2b shows the effect of increasing number of classifier-sets(N), N is a parameter which controls both efficacy and speed. We observe that as N increases, the percentages error initially decreases and then slowly increases The speed of the algorithm also decreases. From our experiments

Algorithm 2 Partition based algorithm

– Training

Goal: To build N classifier-sets
Input: M Training images, $S \leftarrow \emptyset$
for $k = 1$ to N **do**
    Partition training images into k^2 parts, $C \leftarrow \emptyset$
    for $p = 1$ to k^2 **do**
        Train a Classifier on p^{th} partition over all training images, call it KF
        $C \leftarrow C \cup \{KF\}$
    end for{ Now $C = \{KF_1, KF_2, ...KF_{k^2}\}$ }
    $S \leftarrow S \cup \{C\}$
end for{ Now S contains $\{C_1, C_2, ...C_N\}$ }

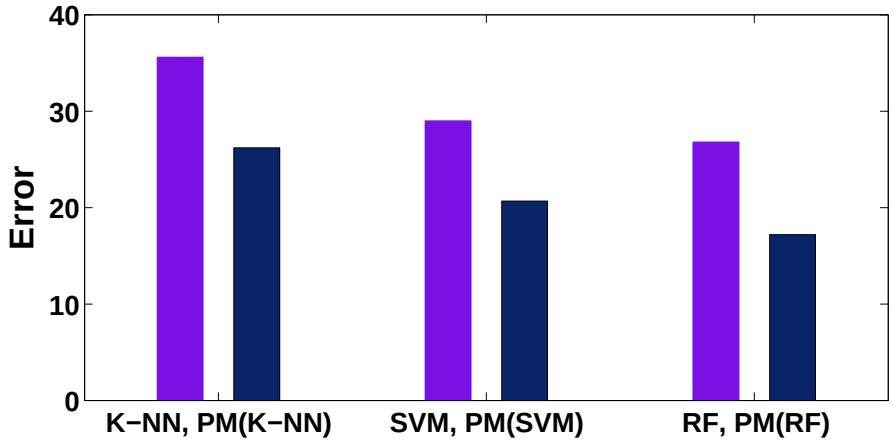
– Characterize Terrain of given image

Input: Image I
for all patches of Image I **do**
    $L \leftarrow \emptyset$
    for $i = 1$ to N **do**
        $l \leftarrow$ get the label of the patch from classifier set C_i
        $L \leftarrow L \cup \{l\}$
    end for
    Majority voted label from L is declared as final label of the patch.
end for

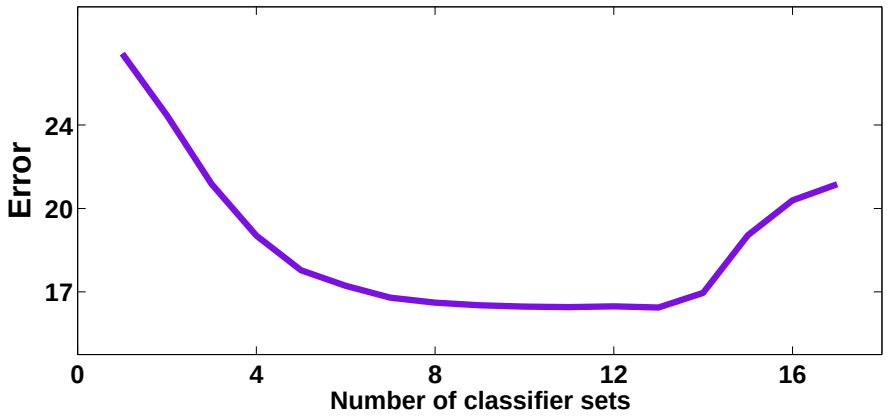
we found that, the optimal choice for N is 5, which has high efficacy and without compromising speed.

Dataset	RF	PM	RF	PM	AVG	Err
O	26.8	17.2	08.7	01.0	35.5	05.6
P-A	18.2	07.9	06.9	00.6	42.3	04.3
P-B	18.9	06.8	05.2	00.4	45.1	04.3

Table 5.1: 1^{st} and 2^{nd} column represents percentage errors of RandomForest(RF) and our partition based algorithm(PM). 3^{rd} and 4^{th} column represents smoothness-error, which corresponds to experiment-3. 5^{th} and 6^{th} column represents the percentage of images, that were labelled just by using Temporal-label-transfer method in Section 5.4, where AVG: Average of percentages of portion of labels that are transferred over sequence of 100 images and Err: Error in label transfer



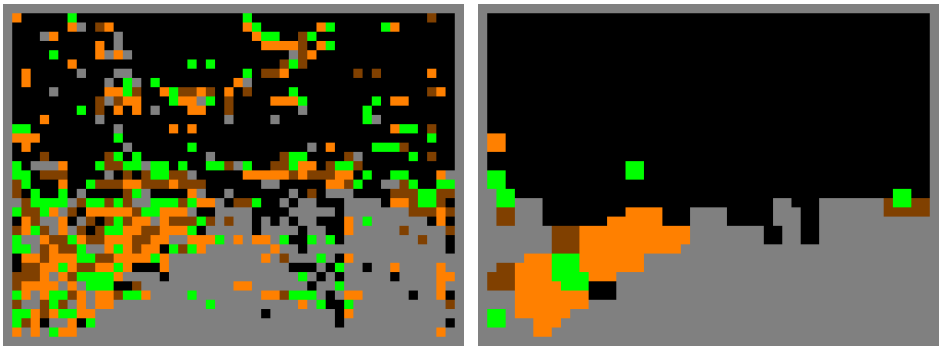
(a)



(b)

Figure 5.2: (a) Comparison of base-line classifiers with Partition-based algorithm operated over them. (b)Error rates by using multiple classifier sets.

(a)



(b) (c)

Figure 5.3: (a) Test image (b) Characterization by RF classifier (c) Characterization by Partition based method

5.3.3 Experiment 3: Spatial smoothness test

In Table 5.1, third and fourth columns show the smoothness-errors of RF and PM operated on RF(PM_RF), on three datasets. Smoothness-error is the difference between percentage errors before and after applying smoothing algorithm (MRF [122]) on the predicted labelled image. We observe that our algorithm has a negligible smoothness-error compared to RFs, which clearly shows that PM_RF itself is capable of characterizing the image smoothly in spatial context. Figure 5.3 shows the superiority of partition based algorithm over baseline RF classifier. We observe that the images labelled using our method are smooth in spatial context.

5.3.4 Discussion

Figure 5.4 shows sample test cases of the Partition based algorithm. We observe that all the classifications are smooth in spatial context. The predicted output in Figure 5.4a and Figure 5.4b are

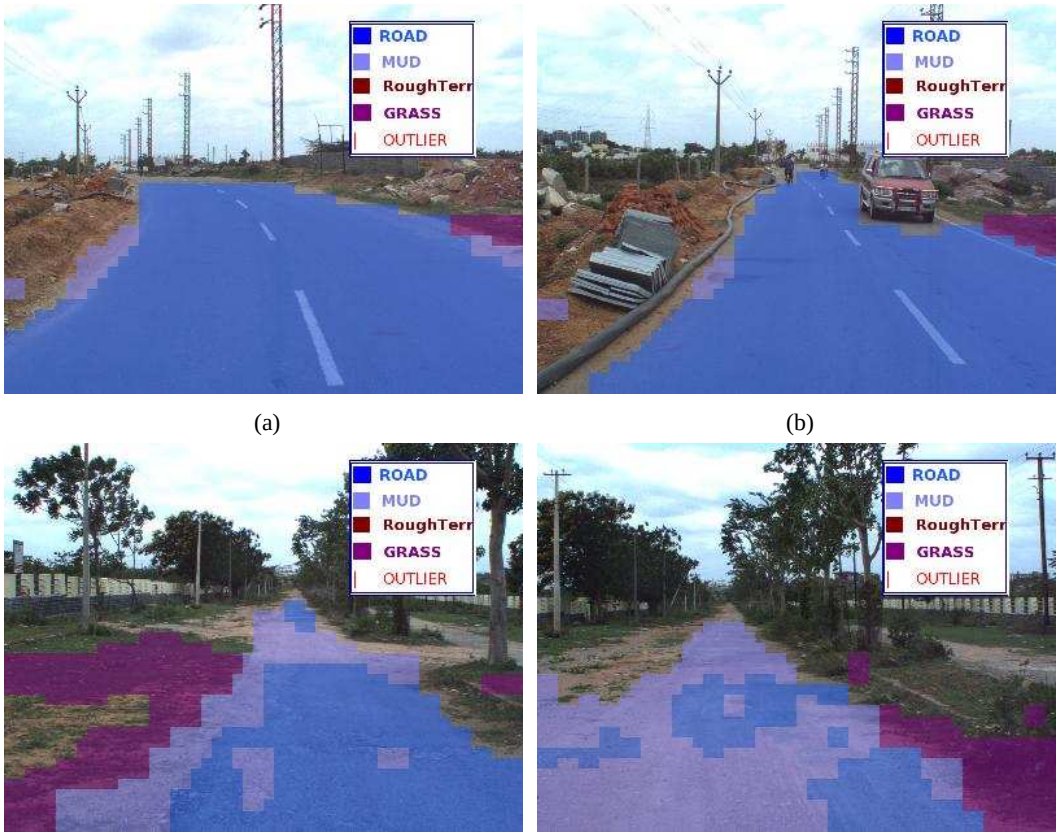


Figure 5.4: Test images blended with predicted classifications

appreciable, as we can see, even the moving vehicle was not classified as traversible path. From Figure 5.4c and Figure 5.4d we observe that there are big trees present in the scene, which have a very similar characteristics of grass, yet they were classified as “Other” class, this was mainly due to the efficient learning of relationship between positions of different classes in the Parition-based-algorithm. The classification on the left side of the Figure 5.4c is incomplete, this is mainly because the patches have the characteristics of both grass and mud, these patches look like mud when seen from far and confusion arises as the camera gets closer to these patches. These patches can be classified correctly if one uses the temporal classification information. In the following section 5.5 we introduce our Adaptive method to handle these problems.

5.4 From Image to Video

Temporal label transfer. Most of the methods in literature deal with single image. They do not use the fact that they are dealing with a sequence of continuous video stream. When robot navigates through terrain, the camera captures sequence of frames. Any two consecutive frames have lot of common image regions. In order to characterize the terrain of the image using traditional machine learning based algorithm some kind of feature is extracted from each patch. The feature vector is fed to a classifier, which returns the label of the patch. Note that in this process, feature extraction is computationally expensive. In our case, when a new frame is captured by the camera, fast coarse optical flow [123] between the previously captured frame and current frame is calculated. For each patch of the new frame, if there is flow present, we transfer the corresponding label from the previous frame to the current frame, else feature is extracted from the patch and fed to our partition-based algorithm. In this way without even extracting features from the current frame, we can label considerable portion of the frame.

We conduct an experiment to see, what portion of the image can be labelled by just using temporal label transfer. The average percentage of image that is labelled correctly over testing images is reported in fifth and sixth column of Table 5.1. We observed that by just using temporal label transfer, we can label approximately 40% of the image on three datasets with significantly lower percentage error. This saves around 40% of the total time taken (which includes feature extraction time and classification time), such a reduction in time is crucial in real time systems like robots.

5.5 Adaptive method

The canonical offline or memory-less classifiers tend to perform poorly in outdoor environments because these environments contain huge variations in illumination. One of the solutions to this

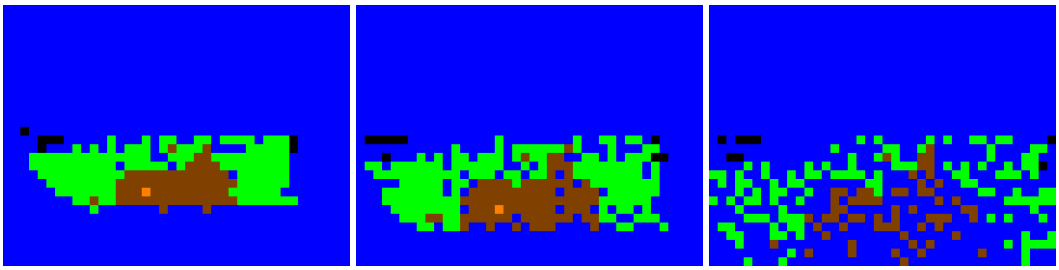


Figure 5.5: Tracked patch-labels across three frames.

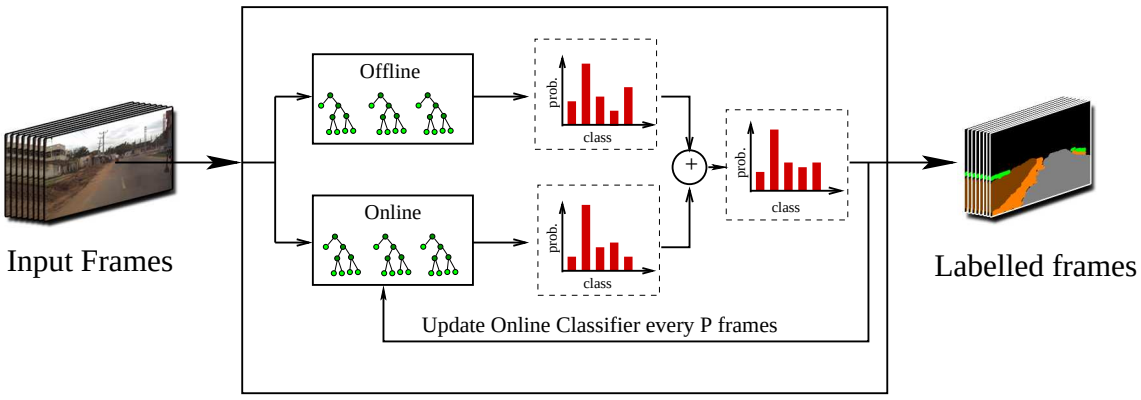


Figure 5.6: Block diagram of the proposed scheme

Algorithm 3 Adaptive algorithm

– Training

$I_c \leftarrow$ current image that needs to be classified.
$P \leftarrow$ Number of previous frames to use.
$stepSize \leftarrow$ Number of frames from which the patches are to be tracked.
$previousFrames = \{I_{c-P}, I_{c-P+1}, I_{c-P+2}, ...I_c\}$
$newTrainingData \leftarrow \emptyset$
for $i = 1$ to $\lfloor P/stepSize \rfloor$ **do**
    $j = c - i * stepSize$
    Track patches from the previous frames $\{I_{j+1}, I_{j+2}, ...I_{j+stepSize}\}$.
    for all tracked patches p_j **do**
        $Label(p_j) \leftarrow$ {Most repeating label among $stepSize$ labels}
    end for
    Update $newTrainingData$ with tracked patches and their corresponding labels.
end for
$onlinePMmodel \leftarrow$ Get the model from Partition-based-method trained on $newTrainingData$.

– Characterize Terrain of given image

Input: Image I_c
for all patches of Image I_c **do**
    $P1 \leftarrow$ the posterior probabilities from offlinePMmodel
    $P2 \leftarrow$ the posterior probabilities from onlinePMmodel
    $P1 = P1 + P2$ {Fuse the results}
    Label corresponding to maximum probability in P is declared as final label of the patch
end for

problem is to train the algorithm on all possible variations of illuminations, which is impractical.Also, In general, increasing the amount of training data drastically, decreases the performance of classifier. These motivate us for developing a terrain classification scheme, that is capable of classifying the terrain in dynamic environments. Previous laser based solutions [4] for this problem are appreciable, but our aim is to classify terrain using only monocular camera, where collecting online ground-truth is impossible. In this section, we describe our scheme for this problem that would enable the robot to adapt to unseen images.

In the proposed algorithm (summarized in Algorithm 3 in page 71), let us denote the current frame with I_i. The previous P frames would be I_{i-P},I_{i-P-1}...I_{i-1}, which are already labelled by our scheme. Using the previously computed flow between successive frames in Section 5.4, we track the patches from previous frames at an interval of K frames. We use $K = 5$ in our experiments. For example, Figure 5.5 shows the tracked patches from three successive frames. These tracked patches across frames slowly vary in their illumination and perspectivity. For each of the tracked patches, we have K labels associated from the K frames. We label the tracked patches accurately by selecting the most repeating label from the K labels. We train another Partition-based classifier on these tracked patches on previous P frames, we call this classifier as online-partition-based-classifier. We update the online-partition-based-classifier every P frames.

To characterize the terrain of the current frame, the posterior probabilities of Offline partition based classifier and online-partition-based-classifier are added (see Figure 5.6 in page 70). In case, two of the posterior probabilities are close, we choose the label that is most repeated with in the neighborhood of the patch.

Implementation details: In our experiments we train online-partition-based-classifier every 200 frames, note that while training the captured frames are classified independently. Hence online training and classification can be executed in parallel. Also using RFs internally adds another advantage. In RF, the final posterior probability is fused result of several posterior probabilities of several trees, here each tree can be used independently and hence can be executed in parallel. These advantages make our algorithm parallel and can be implemented efficiently using GPUs [124].

5.5.1 Performance Gain Due to Adaptive Method

In this section we show both by quantitative and qualitative experimental results the advantages of having an online classifier. Quantitatively we show decrease in errors on 6 data sets, including two publicly available data sets. Qualitatively we show those portions in the image where the adaptive classifier has corrected wrongly classified patches by the Partition method. We also show results from an experimental run where the vehicle reaches the location from where it started its journey.

Dataset	PM	Adaptive	Error-rate-reduction
O-A	18.2	12.7	30.2
O-B	20.2	15.9	21.2
O-C	17.0	13.3	21.7
O-D	17.5	16.1	07.9
P-A	07.9	05.3	32.9
P-B	06.8	06.1	10.2

Table 5.2: Comparison of Adaptive algorithm with Offline-partition-based-method

Quantitative and qualitative results

Table 5.2 shows the percentage errors of Offline-partition-based-method and Adaptive algorithm on 6 sequences in columns 2 and 3. Since the Offline-partition-based-method already achieves a reasonably low percentage errors, further improvements over Offline-partition-based-method by Adaptive algorithm can be portrayed in terms of rate of decrease in error, which is given by

$$Error\,rate\,reduction = \frac{\%\,error\,of\,PM\;-\;\%\,error\,of\,Adaptive\,algorithm}{\%\,error\,of\,PM} \tag{5.1}$$

The error-rates were presented in column 4 of Table 5.2. The first four rows of the table correspond to 4 sequences of our dataset. In these sequences, the robot is navigated continuously until 800 frames were captured. Adaptive algorithm is applied on these 4 sequences independently, where the online-classifier is updated every 100 frames. The last two rows show the percentage errors on datasets by Procopio [4], since their data-set is a sequence of only 100 frames, the online-classifier is updated every 20 frames. 20 randomly picked images from each sequence were used for testing. We observe that the Adaptive algorithm has a huge decrease in error-rate of more than 20% on almost all the sequences. This clearly shows the superiority of the proposed scheme.

Figure 5.7 shows some of the test images marked with the red-colored-patches from our dataset. They represent the labels that are correctly labelled by Adaptive algorithm, which are wrongly labelled by offline Partition-method.

Closed loop test

The closed loop test is a means to evaluate if the performance of the adaptive algorithm improves over time, the knowledge embedded in the classifier is not static and has adapted with passage of time. The improved performance comes by exploiting the data that comes on the fly, while

Figure 5.7: Test images marked with red-colored-patches, representing the labels that are correctly labelled by Adaptive algorithm but wrongly labelled by offline Partition method.

simultaneously not forgetting what was learned at bootstrap. At the beginning of the run the robot has learned based on the offline dataset representing bootstrapped knowledge. As the run progresses the knowledge is expected to be enhanced. By showing improved performance upon reaching the starting location after a run of more than 2km we verify that the objective of learning without forgetting the past is realized.

In this experiment, we test our Adaptive algorithm in a closed loop path (see Figure 5.8 in page 75)i.e., the Adaptive algorithm is applied on data which was collected by navigating the robot on the same road twice. 20 random images from each loop at approximately same locations were used for testing. Note that not even one of these images were used in the initial offline training dataset. We observe that the mean error on the round-1 is 16%, where as the mean error for round-2 was observed to be 13%. The decrease in percentage error was observed mainly because, the adaptive algorithm slowly adapts itself to the new environments. Second row of the Figure 5.8 shows the test image along with the predicted labelled images from the first and second loops. We observe

Figure 5.8: 1_{st} row: Path navigated by robot in a closed loop, marked in green color. 2_{nd} row: Test image with predicted labelled images from the first and second loops.

that the wrongly labelled mud(orange) patches in first loop are being correctly labelled in the second loop.

5.6 Discussion

This chapter presented a novel partition-based algorithm for classification of outdoor terrains using monocular camera. The partition-based algorithm is fast as it is build on top of Random Forests. Three experiments were conducted verifying different aspects of the algorithm. The proposed algorithm is generic and enhanced the percentage error of base-line classifiers by approximately 10%. The partition-based algorithm was extensively tested on our dataset and on other publicly available datasets and its efficacy established.

Partition-based method was extended to Adaptive algorithm by learning from the data by fruitfully exploiting the data that was obtained on the fly. Concepts may drift over time, offline classifiers may not adapt to these drift as effectively as a classifier that also adapts online. The adaptive algorithm was tested on several data sets, where an average decrease in error rate of around 20% was observed to portray its advantages. Further we show results where a vehicle upon coming back to the same starting point after traversing a loop of more than 2km improves its performance during the second traversal of the loop. This demonstrates that the adaptive classifier is able to adapt to changes that occur during a traversal while holding on to what was learned at bootstrap or before the commencement of navigation. The future scope of our work includes much better processing of the video data using complex temporal clues along with fusing geometric and appearance clues.

Chapter 6

Conclusions and Future Work

We have put forth new techniques in two different areas of scene interpretation namely, scene reconstruction in computer vision and scene recognition in mobile robotics. Our proposed framework deals with reconstruction of piecewise planar scenes from videos in much the same way as Bundle Adjustment for point sets. Multiple planes and views are taken into consideration and algorithm does not impose the constraint that all the planes should be visible in a single view. Furthermore, the presence of inter-image homographies present useful robustness to outliers, that may not have been pruned in the initial stages of registration and homography computation. This makes it a useful "bridge" between initialization approaches and non-linear minimization methods.

Next we addressed the problem of scene recognition in mobile robotics. Due to unavailability of existing datasets for experiments and comparisons, we prepared our own dataset. Our annotated dataset comprises of outdoor rural and urban terrains, which contain huge varieties of scenes under varied environmental conditions. We have reported extensive comparison of various classifiers operating on features for classification of outdoor terrains using only monocular camera. We have analysed the performance of different classifiers and studied the effect of various parameters such as the richness of the features and the patch size on the classifier performance. We reported that Random forests trained on weighted color cum texture feature gives the best baseline result, with an error of 25.5% compared to other classifiers such as Naive Bayes, Artificial neural networks, K-nearest neighbors and Support vector machines. On other publicly available dataset the baseline error rate was 18.2%. We conducted various empirical studies with state-of-the-art machine learning techniques and various parameters of the problem.

We presented a novel and fast partition-based algorithm for classification of outdoor terrains using monocular camera. The speed of partition-based algorithm is attributed to Random Forests on top of which the algorithm is built. Several experiments were conducted to ascertain the usability

of our method. Our algorithm is generic and has reduced the percentage error of base-line Random Forest classifier by approximately 10%. We have tested our partition-based algorithm extensively on our dataset and on other publicly available datasets to validate its efficacy. We optimize the performance of our algorithm by limiting it to regions where the temporal label transfer is not applicable.

Partition-based method was extended to Adaptive algorithm by learning from the data that was obtained on the fly. The adaptive algorithm was tested on several data sets and an average decrease in error rate of around 20% demonstrated its advantages. We also conducted experiments in which a vehicle comes back to the same starting point after traversing a loop of more than 2km. An improved performance is observed during the second traversal of the loop. This indicates that the adaptive classifier is capable of adapting to changes that occur during a traversal while retaining what was learned at bootstrap or before the commencement of navigation.

6.1 Future work

The proposed framework consists of quasi-convex objective functions, though quasi-convex problems have a guaranteed optimal solution, they are iterative in nature. One could investigate in designing convex objective functions, which would have an advantage of non-iterative(fast) solutions, making them much suitable for faster computation required by videos. The proposed objective functions show robustness to noise, they may be still made much robust using existing literature on convex optimization [114] One important concern with any algorithm is its ability to handle outliers. Currently our algorithm handles only the noise in the data, the existing framework could be extended to handle outliers with L_∞ norm using convex formulations. This investigation may also help in solving other problems of geometric vision. Investigating the design of a hybrid algorithm which is based on objective functions from both Bundle adjustment and Convex optimization frameworks also stands as important extension. Recent literature on Practical Global Optimization [105, 115] may be utilized to improve the running time of our algorithm. We however believe that our work lays down new and important directions for the problem of planar reconstruction.

In partition-based algorithm, currently the output predictions from different classifier sets are fused by using simple statistical mode operator. This could be enhanced by using weighted-map for each classifier set followed by integrating the results from each classifier set. Also we could decrease the computational time of the partition-based algorithm by using the classifier sets dynamically, *i.e.*, one could use few classifier sets to start and use other classifier sets only if the predicted labels are different. The Adaptive algorithm could be enhanced using the newly introduced semi-supervised machine learning techniques especially the semi-supervised Random Forests. We could process the

video data using complex temporal clues and then integrate geometric and appearance clues in an optimization framework.

Related Publications

The following publications resulted from work in and related to this thesis

- Visesh Chari, Anil Nelakanti, Chetan Jakkoju and C. V. Jawahar. “Piecewise Planar Reconstruction using Convex Optimization.” *In proceedings of Asian Conference on Computer Vision (ACCV'09).*
- Chetan J., Madhava Krishna and C. V. Jawahar. “Fast and Spatially-smooth Terrain Classication using Monocular Camera.” *In proceedings of International Conference on Pattern Recognition. (ICPR 2010)*
- Chetan J., Madhava Krishna and C. V. Jawahar. “An Adaptive Outdoor Terrain Classification Methodology using Monocular Camera” *In proceedings of International Conference on Intelligent Robots and Systems. (IROS 2010)*

Bibliography

[1] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2004.

[2] Q. Ke, "Robust subspace approach to extracting layers from image sequences," in *Phd Thesis*, 2003, pp. 6–9.

[3] C. Weiss, H. Frohlich, and A. Zell, "Vibration-based terrain classification using support vector machines." in *International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China., October 2006, pp. 4429–4434.

[4] M. J. Procopio, J. Mulligan, and G. Grudic, "Learning in dynamic environments with ensemble selection for autonomous outdoor robot navigation," in *International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[5] C. Baillard and A. Zisserman, "Automatic reconstruction of piecewise planar models from multiple views." in *Computer Vision and Pattern Recognition(CVPR)*, Fort Collins, Colorado, USA, June 1999, pp. 559–565.

[6] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *ACM SIGGraph, Computer Graphics*, 1996.

[7] A. R. Dick, P. H. S. Torr, S. J. Ruffle, and R. Cipolla, "Combining single view recognition and multiple view stereo for architectural scenes," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2001, pp. I: 268–274.

[8] A. W. Fitzgibbon, G. Cross, and A. Zisserman, "Automatic 3D model construction for turntable sequences," in *3D Structure from Multiple Images of Large-Scale Environments*, 1998.

[9] P. Gargallo and P. F. Sturm, "Bayesian 3D modeling from images using multiple depth maps," in *Computer Vision and Pattern Recognition(CVPR)*, 2005, pp. II: 885–891.

[10] H. L. Jin, S. Soatto, and A. J. Yezzi, "Multi-view stereo reconstruction of dense shape and complex appearance," *International Journal of Computer Vision(IJCV)*, vol. 63, no. 3, pp. 175–189, July 2005.

[11] K. N. Kutulakos, "Approximate N-view stereo," in *Proceedings of the European Conference on Computer Vision(ECCV)*, 2000, pp. I: 67–83.

[12] S. M. Seitz and K. N. Kutulakos, "A theory of shape by space carving," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 1999, pp. 307–314.

[13] F. Lang and W. Forstner, "3D-city modeling with a digital one-eye stereo system," in *In Proceedings of the XVIII ISPRS-Congress.*, Vienna, Austria, July 1996.

[14] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *Pattern Analysis and Machine Intelligence(PAMI)*, vol. 27, no. 3, pp. 418–433, 2005.

[15] D. Morris and T. Kanade, "Image-consistent surface triangulation," in *Computer Vision and Pattern Recognition(CVPR)*. Los Alamitos: IEEE, jun 2000, pp. 332–338.

[16] A. Nakatuji, Y. Sugaya, and K. Kanatani, "Optimizing a triangular mesh for shape reconstruction from images," *Transactions Institute Elec. Info. and Comm. Eng.*, vol. E88-D, no. 10, pp. 2269–2276, Oct. 2005.

[17] D. Scharstein and R. S. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision(IJCV)*, vol. 47, no. 1-3, pp. 7–42, Apr. 2002.

[18] C. Strecha, R. Fransens, and L. J. V. Gool, "Wide-baseline stereo from multiple views: A probabilistic account," in *Computer Vision and Pattern Recognition(CVPR)*, 2004, pp. I: 552–559.

[19] C. Vestri and F. Devernay, "Using robust methods for automatic extraction of buildings," in *Computer Vision and Pattern Recognition(CVPR)*. IEEE Computer Society, 2001.

[20] A. E. Bartoli and P. F. Sturm, "Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene," *International Journal of Computer Vision(IJCV)*, vol. 52, no. 1, pp. 45–64, Apr. 2003.

[21] A. Streilein and U. Hirschberg., "Integration of digital photogrammetry and caad: Constraint-based modeling and semi-automatic measurement," in *In Proceedings of the International CAAD Futures Conference*, Singapore, September 1995.

[22] P. A. Beardsley, P. H. S. Torr, and A. Zisserman, "3D model acquisition from extended image sequences," in *Proceedings of the European Conference on Computer Vision(ECCV)*, 1996, pp. II:683–695.

[23] M. Pollefeys, M. Vergauwen, , and L. V. Gool., "Automatic 3d modeling from image sequences," in *In Proceedings of the XIX ISPRS-Congress*, July 2000, pp. I: 619–626.

[24] N. S. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys, "Interactive 3d architectural modeling from unordered photo collections," *ACM Trans. Graph.*, vol. 27, no. 5, p. 159, 2008.

[25] H. S. P. Torr, R. Szeliski, and P. Anandan, "An integrated bayesian approach to layer extraction from image sequences," *Pattern Analysis and Machine Intelligence(PAMI)*, vol. 23, no. 3, pp. 297–303, 2001.

[26] U. Frese, "A discussion of simultaneous localization and mapping," *Auton. Robots*, vol. 20, no. 1, pp. 25–42, 2006.

[27] A. H. A. Hafez, S. Bhuvanagiri, K. M. Krishna, and C. V. Jawahar, "On-line convex optimization based solution for mapping in VSLAM," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2008, pp. 4072–4077.

[28] N. Karlsson, E. D. Bernardo, J. P. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vSLAM algorithm for robust localization and mapping," in *Proceedings of International Conference on Robotics and Automation(ICRA)*. IEEE, 2005, pp. 24–29.

[29] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," in *International Conference on Intelligent Robots and Systems (IROS)*, 2005.

[30] P. Vernaza, B. Taskar, and D. D. Lee., "Online self-supervised terrain classification via discriminatively trained submodular markov random fields." in *Proceedings of International Conference on Robotics and Automation(ICRA)*, 2008.

[31] D. Sadhukhan, "Autonomous ground vehicle terrain classification using internal sensors." in *Masters thesis, Dept. Mech. Eng.,*, vol. 21, Florida State University, Tallahassee, Florida, USA., 2004, pp. 1185–1191.

[32] C. Brooks and K. Iagnemma, “Vibration-based terrain classification for planetary exploration rovers.” in *IEEE Transactions on Robotics*, vol. 21, 2005, pp. 1185–1191.

[33] M. R. Blas, M. Agrawal, A. Sundaresan, and K. Konolige, “Fast color/texture segmentation for outdoor robots,” in *International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[34] C. Rasmussen, “Combining laser range, color and texture cues for autonomous road following,” in *Proceedings of International Conference on Robotics and Automation(ICRA)*, 2002.

[35] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski, “Self-supervised monocular road detection in desert terrain,” in *Robotics: Science and Systems (RSS)*, 2006.

[36] J. K. Y. Ma, S. Soatto, and S. Sastry., “An invitation to 3d,” in *Springer, Berlin*, 2003.

[37] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” in *International Journal on Computer Vision(IJCV)*, 2004, pp. 91–110.

[38] D. Capel, “Image mosaicing and super-resolution,” in *University of Oxford*. PhD Thesis, 2001.

[39] F. Fraundorfer and H. Bischof, “Affine invariant region matching using geometric hashing of line structures,” in *Proceedings of the 27th Workshop of the Austrian Association for Pattern Recognition*, Laxenburg, Austria, 2003, pp. 57–64.

[40] K. Fredrik, “Multiple view geometry and the l-infinity norm,” in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2005.

[41] M. Brown and D. Lowe., “Recognising panoramas,” in *Proceedings of the Ninth International Conference on Computer Vision.*, Nice, France, 2003, pp. 1218–1225.

[42] P. Pritchett and A. Zisserman., “Wide baseline stereo matching,” in *Proceedings of the International Conference on Computer Vision (ICCV)*, Bombay, India., 1998, pp. 754–760.

[43] M. Lourakis, S. Halkidis, and S. Orphanoudakis., “Matching disparate views of planar surfaces using projective invariants.” in *Image and Vision Computing.*, 2000, pp. 673–683.

[44] M. Lourakis, S. Tzurbakis, A. Argyros, and S. Orphanoudakis., “Feature transfer and matching in disparate stereo views through the use of plane homographies.” in *Pattern Analysis and Machine Intelligence(PAMI)*, 2003, pp. 271–276.

[45] T. Park, S. Fleishman, D. Cohen-Or, and D. Lischinski., "Compression of indoor video sequences using homography-based segmentation." in *Proceedings of the Pacific Conference on Computer Graphics and Applications.*, 2000, pp. 290–299.

[46] R. Szeliski and P. Torr, "Geometrically constrained structure from motion: points on planes." in *Proceedings of the SMILE98-3D Structure from Multiple Images of Large-Scale Environments.*, 1998, pp. 171–186.

[47] R. Kaucic, R. Hartley, and N. Dano, "Plane-based projective reconstruction." in *Proceedings of the International Conference on Computer Vision (ICCV)*, Vancouver, Canada, 2001.

[48] C. Rother and S. Carlsson., "Linear multiview reconstruction and camera recovery." in *Proceedings of the International Conference on Computer Vision (ICCV)*, Vancouver, Canada, 2001, pp. 42–51.

[49] R. Kumar, P. Anandan, and K. Hanna., "Shape recovery from multiple views: a parallax based approach." in *DARPA Image Understanding Workshop.*, Monterrey, CA, 1994.

[50] M. Irani, P. Anandan, and D. Weinshall., "From reference frames to reference planes: Multiview parallax geometry and applications." in *Proceedings of the European Conference on Computer Vision(ECCV)*, Freiburg, Germany, 1998, pp. 829–845.

[51] O. D. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," in *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, vol. 2, June 1988, pp. 485–508.

[52] Z. Zhang and A. R. Hanson, "3d reconstruction based on homography mapping." in *ARPA Image Understanding Workshop*, 1996.

[53] T. Sanger, "Stereo disparity computation using gabor filters," *Biological Cybernetics*, vol. 59, pp. 405–418, 1988.

[54] A. A. Tony Jebara and A. Pentland, "3d and stereoscopic visual communication," in *IEEE Signal Image Processing*, 1999.

[55] T. Huang and A. Netravali, "Motion and structure from feature correspondences: A review," in *Proceedings of IEEE*, 1994.

[56] U. Dhond and J. Aggarwal, "Structure from stereo - a review," in *IEEE Transactions on Systems, Man and Cybernetics*, 1989.

[57] R. Mohr and B. Triggs, "Projective geometry for image analysis," in *Technical report, International Society for Photogrammetry and Remote Sensing, Vienna Congress*, July 1996.

[58] O. Faugeras, "Three-dimensional computer vision: A geometric viewpoint," in *MIT Press*, 1993.

[59] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *Proceedings of the International Workshop on Vision Algorithms*, vol. 1883, 1999, pp. 298–372.

[60] J. Wang and E. Adelson, "Layered representation for motion analysis," in *Computer Vision and Pattern Recognition(CVPR)*, 1993.

[61] A. Jepson and M. Black, "Mixture models for optical flow computation," in *Computer Vision and Pattern Recognition(CVPR)*, 1993.

[62] S. Hsu, P. Anandan, and S. Peleg, "Accurate computation of optical flow by using layered motion representations," in *ICPR*, 1994.

[63] A. N. Tikhonov and V. A. Arsenin, "Solutions of ill-posed problems," in *Winston and Sons*, 1977.

[64] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, 1981, pp. 674–679.

[65] T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature*, vol. 317, pp. 314–319, 1985.

[66] T. Poggio and F. Girosi, "A theory of networks for approximation and learning," in *Technical Report AIM-1140, MIT AI Lab*, 1989.

[67] Q. Ke and T. Kanade, "Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection," in *Computer Vision and Pattern Recognition (CVPR)*, June 2003.

[68] S. Ayer and H. Sawhney., "Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding." in *Proceedings of the International Conference on Computer Vision (ICCV)*, 1995.

[69] Y. Weiss and E. Adelson, "A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models," in *Computer Vision and Pattern Recognition (CVPR)*, 1996.

[70] Y. Weiss, “Smoothness in layers: Motion segmentation using nonparametric mixture estimation,” in *Computer Vision and Pattern Recognition (CVPR)*, 1997.

[71] P. Torr, R. Szeliski, and P. Anandan, “An integrated bayesian approach to layer extraction from image sequences.” in *Proceedings of the International Conference on Computer Vision (ICCV)*, 1999.

[72] J. Rissanen, “A universal prior for integers and estimation by minimum description length,” in *The Annals of Statistics*, 1983, pp. 416–431.

[73] J. Shi and J. Malik, “Motion segmentation and tracking using normalized cuts,” in *Proceedings of the International Conference on Computer Vision (ICCV)*, 1998.

[74] M. Irani, B. Rousso, and S. Peleg, “Detecting and tracking multiple moving objects using temporal integration,” in *Proceedings of the European Conference on Computer Vision(ECCV)*, 1992.

[75] M. J. Black and P. Anandan, “The robust estimation of multiple motions: Affine and piecewise-smooth flow fields,” in *Tr, Xerox PARC*, Dec 1993.

[76] H. Sawhney and S. Ayer, “Compact representations of videos through dominantand mulitple motion estimation,” in *Pattern Analysis and Machine Intelligence(PAMI)*, 1996, pp. 18:814–831.

[77] P. Rousseeuw and A. Leroy, “Robust regression and outlier detection,” in *John Wiley and Sons*, 1987.

[78] M. Black and P. Anandan, “A framework for the robust estimation of optical flow,” in *Proceedings of the International Conference on Computer Vision (ICCV)*, 1993, pp. 231–236.

[79] Y. Altunbasak, P. E. Eren, and A. M. Tekalp, “Region-based parametric motion segmentation using color information,” in *Graphical Models and Image Processing*, January 1998.

[80] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[81] F. Kahl and D. Henrion, “Globally optimal estimates for geometric reconstruction problems,” in *International Journal of Computer Vision*, vol. 74, August 2006, pp. 3–15.

[82] C. Weiss, H. Tamimi, and A. Zell, “A combination of vision- and vibration-based terrain classification,” in *International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, September 2008.

[83] M. A. Sotelo, "Virtuous: Vision-based road transportation for unmanned operation on urban-like scenarios," in *Intelligent Transport Systems*, 2004.

[84] N. Vandapel, D. E. Huber, A. Kapuria, and M. Hebert, "Natural terrain classification using 3-d ladar data," in *Proceedings of International Conference on Robotics and Automation(ICRA)*, 2004.

[85] C. Weiss, N. Fechner, M. Stark, and A. Zell, "Comparison of different approachs to vibration-based terrain classification." in *European Conf. on Mobile Robotics*, 2007.

[86] A. Angelova, L. Matthies, D. Helmick, and P. Perona, "Fast terrain classification using variable-length representation for autonomous navigation," in *Computer Vision and Pattern Recognition(CVPR)*, 2007, pp. 1–8.

[87] L. Ojeda, J. Borenstein, G. Witus, and R. Karlsen, "Terrain characterization and classification with a mobile robot," in *Journal of field robotics*, 2006.

[88] Wolf, D. F. Sukhatme, G. Fox, and D. W. Burgard, "Autonomous terrain mapping and classification using hidden markov models." in *Proceedings of International Conference on Robotics and Automation(ICRA)*, vol. 2, 2005, pp. 2026–2031.

[89] C. Brooks, K. Iagnemma, and S. Dubowsky, "Vibration based terrain analysis for mobile robots." in *Proceedings of International Conference on Robotics and Automation(ICRA)*, 2005.

[90] C. Dima, N. Vandapel, and M. Hebert., "Classifier fusion for outdoor obstacle detection," in *Proceedings of International Conference on Robotics and Automation(ICRA)*, vol. 1, 2004, pp. 665–671.

[91] D. Bradley, R. Unnikrishnan, and J. A. D. Bagnell, "Vegetation detection for driving in complex environments," in *Proceedings of International Conference on Robotics and Automation(ICRA)*, April 2007.

[92] C. Vallespi-Gonzalez and T. Stentz, "Prior data and kernel conditional random fields for obstacle detection," in *Robotics: Science and Systems (RSS)*, Zurich, Switzerland, June 2008.

[93] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," in *Pattern Analysis and Machine Intelligence(PAMI)*, vol. 26, 2004.

[94] M. Nieto and L. Salgado, "Real-time vanishing point estimation in road sequences using adaptive steerable filter banks," in *Advanced Concepts for Intelligent Vision Systems*, 2007.

[95] Y. Alon, A. Ferencz, and A. Shashua, "Off-road path following using region classification and geometric projection constraints," in *the proceedings of Computer vision and Pattern Recognition*, 2006.

[96] M. Pietikainen, T. Nurmela, T. Maenpaa, and M. Turtinen, "View-based recognition of real-world textures," in *Pattern Recognition*, 2004.

[97] M. Varma and A. Zisserman, "Texture classification: Are filter banks necessary ?" in *Computer Vision and Pattern Recognition(CVPR)*, 2003.

[98] J. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin dags for multiclass classification," in *Proceedings of Advanced Neural Information Processing Systems.*, 2000, pp. 547–553.

[99] L. Breiman, "Random forests." in *Machine Learning*, 2001.

[100] A. Jaiantilal, "Random forest implementation in matlab," *Website: http://code.google.com/p/randomforest-matlab/*, May 2009.

[101] R. Hartley and F. Kahl, "Global optimization through searching rotation space and optimal estimation of the essential matrix," in *Proceedings of the International Conference on Computer Vision (ICCV)*, October 2007, pp. 1–8.

[102] V. Chari, A. Nelakanti, C. Jakkoju, and C. V. Jawahar, "Piecewise planar reconstruction using convex optimization," in *Asian Conference on Computer Vision (ACCV'09)*, 2009.

[103] P. H. S. Torr and D. W. Murray, "The development and comparison of robust methodsfor estimating the fundamental matrix." in *International Journal of Computer Vision(IJCV)*, vol. 24(3), 1997, pp. 271–300.

[104] A. Bartoli., "A random sampling strategy for piecewise planar scene segmentation," in *Computer Vision and Image Understanding*, vol. 105(1), January 2007, pp. 42–59.

[105] S. Agarwal, M. Chandraker, F. Kahl, D. Kriegman, and S. Belongie, "Practical global optimization for multiview geometry," in *Proceedings of the European Conference on Computer Vision(ECCV)*, 2006.

[106] M. Salzmann, R. Hartley, and P. Fua, "Convex optimization for deformable surface 3-d tracking." in *Computer Vision and Pattern Recognition(CVPR)*, 2007.

[107] K. Mitra and R. Chellappa, “A scalable projective bundle adjustment algorithm using the l norm,” in *The Indian Conference on Computer Vision, Graphics and Image processing (ICVGIP)*, 2008.

[108] A. Bartoli and P. Sturm, “Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene,” in *International Journal of Computer Vision(IJCV)*, vol. 52(1), April 2003, pp. 45–64.

[109] M. Chandraker and D. Kreigman, “Convex optimization for bilinear problems in computer vision,” in *Computer Vision and Pattern Recognition(CVPR)*, 2008.

[110] A. Criminisi, I. Reid, and A. Zisserman, “A plane measuring device,” *Image and Vision Computing*, vol. 17, no. 8, pp. 625–634, 1999.

[111] J. Weng, T. S. Huang, and N. Ahuja, “Motion and structure from two perspective views: Algorithms, error analysis, and error estimation,” *Pattern Analysis and Machine Intelligence(PAMI)*, vol. 11, no. 5, pp. 451–476, 1989.

[112] H. S. Sawhney, S. Hsu, and R. Kumar, “Robust video mosaicing through topology inference and local to global alignment.” in *Proceedings of the European Conference on Computer Vision(ECCV)*, vol. 2, 1998, pp. 103–119.

[113] C. Olsson, F. Kahl, , and M. Oskarsson, “Optimal estimation of perspective camera pose.” in *Proceedings of the 18th International Conference on Pattern Recognition*, vol. 02, 2006, pp. 5–8.

[114] Q. Ke and T. Kanade, “Quasiconvex optimization for robust geometric reconstruction,” in *Proceedings of the International Conference on Computer Vision (ICCV)*, Beijing, China, October 2005.

[115] S. Agarwal, N. Snavely, and S. Seitz, “Fast algorithms for l-inf problems in multiview geometry,” in *Computer Vision and Pattern Recognition (CVPR)*, 2008.

[116] J. Chetan, M. Krishna, and C. V. Jawahar, “Fast and spatially-smooth terrain classication using monocular camera,” in *International Conference on Pattern Recognition (ICPR’10)*, 2010.

[117] N. Dalal, *INRIA Person Dataset*. Online, 2005. [Online]. Available: http://pascal.inrialpes.fr/data/human/

[118] P. Arbelaez, C. C. Fowlkes, and D. R. Martin, *The Berkeley Segmentation Dataset and Benchmark*. Online, 2007. [Online]. Available: http://www.cs.berkeley.edu/projects/vision/grouping/segbench/

[119] G. Brostow and J. Fauqueur, "Interactlabeler 1.2.1," *Website: http://mi.eng.cam.ac.uk/projects/cvgroup/software/index.html*, February 2007.

[120] D. Kim, S. M. Oh, and J. M. Rehg, "Traversability classification for ugv navigation: A comparison of patch and superpixel representations," in *Proceedings of International Conference on Robotics and Automation(ICRA)*, 2007.

[121] J. Chetan, M. Krishna, and C. V. Jawahar, "An adaptive outdoor terrain classification methodology using monocular camera," in *International Conference on Intelligent Robots and Systems (IROS'10)*, 2010.

[122] Z. Kato, T. C. Pong, and J. C. M. Lee., "Color image segmentation and parameter estimation in a markovian framework," in *Pattern Recognition Letters*, 2001.

[123] H. Liu, R. Chellappa, and A. Rosenfeld, "Fast two-frame multiscale dense optical flow estimation using discrete wavelet filters." in *Journal of the Optical Society of America*, vol. 20(8), August 2003, pp. 1505–1515.

[124] H. Nguyen, Ed., *GPU Gems 3*. Addison Wesley, 2007.