UNDERSTANDING SCENE STRUCTURE FROM IMAGES

BY

MARIYAM KHALID

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Adviser:

Assistant Professor Svetlana Lazebnik

ABSTRACT

The task of recovering 3D information from 2D images has long been a focus of Computer Vision research. Such information is useful in many applications: from robot navigation, where it allows the robot to understand the physical constraints of the environment it is in, to augmented reality, where 3D information is used to alter images and videos in physically plausible ways. While much progress has been made in this line of research there is still scope for further improvement. This is especially true in the case of pictures taken "in the wild", where variables such as the presence of clutter, people, irregularly shaped buildings, unusual camera angles, etc tend to cause current techniques to fail.

In this work we focus on recovering 3D information from images in the presence of clutter and other such variables. We work on both indoor and outdoor scenes, utilizing different approaches in each case in order to make the 3D information recovery more robust.

Since this work focuses on expanding existing techniques to work well on more challenging datasets, we had to create new datasets for both indoor and outdoor scenes that could test the robustness of our methods. Details of these datasets are also provided in this work.

For Ami and Abu. Always.

ACKNOWLEDGMENTS

I would like to thank the UIUC Computer Vision family for supporting me throughout my degree. My advisor (Dr. Svetlana Lazebnik), Professors (Dr. Derek Hoiem and Dr. David Forsyth) and fellow grad students (Daphne, Kevin, Alice, Jason, Kevin, Cecilia, Bryan, Kevin and Saurabh) have made my graduate school experience one of the most challenging yet rewarding phases of my life.

I would like to thank my family: Ami, Abu, Usman, Sarah and Saim. Thanks to you I feel safe and loved no matter where I am or where you guys are. I hope I stop making you worry as much as I do, even though I know you enjoy my stories.

Finally, I would like to thank my UIUC family: Zainab, for always rushing to my defence, Ahmad, for overkilling it with his concern, Zain, for reminding me to always trust my judgement, Nayab, for her child-like enthusiasm and Shayan, for the safe-walks and the safe-walk talks.

TABLE OF CONTENTS

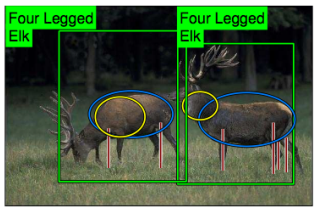
CHAPTER 1

INTRODUCTION AND OVERVIEW

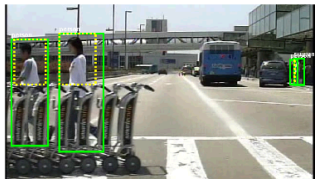
In the last century photography has emerged as one of the primary methods through which humans document and analyze the world around them. Advances in photography have altered the way we live our lives. The way we communicate with each other, our understanding of the world around us, our insights on socio-political events, ... these are just some of the aspects of our lives revolutionized by modern imaging technology.

From a computational point of view, the development of imaging technology has provided an additional signal which can be fed to computers to process and analyze. Computer Vision, the branch of computer science that deals systems that take images as inputs and derive information from them, has actively been working on utilizing and developing techniques from other areas such as signal processing and machine learning and applying them to 2D image inputs to derive useful information. This has lead to the development of techniques such as object detection, face recognition, pose estimation (among many others) which have already been integrated into systems we use in our everyday lives (see figure 1.1).

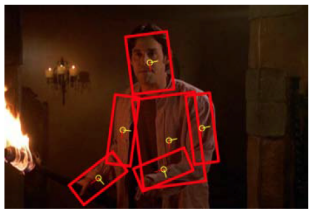
In this thesis we focus on using images of both indoor and outdoor scenes



(a) Object detection.Sample from results produced by Endres et al.[1].



(b) Pedestrian tracking. Sample from results produced by Dollar et al. [2].



(c) Pose estimation. Sample from results produced by Andriluka et al.[3].

Figure 1.1: Examples of computer vision techniques being applied to real-world problems.

Figure 1.2: Example image showing 3-D information being lost in an image projection. Photo taken from [1]

as our inputs and automatically predicting the 3-D structure of the scene. Since an image is a 2-D projection of the 3-D world, a lot of 3-D information is lost when we take a picture of a scene. This phenomenon can be observed in figure 1.2 where there is no observable difference between the toy car and the real car. In fact if it hadn't been for the hand holding the toy car it would have been almost impossible for even a human to tell the difference between the two. Such differences are caused by a phenomenon known as perspective distortion which will be explained in chapter 2. Thus the task of reprojecting 2-D information into the 3-D space, or at least some subset of the 3-D space, is one in which there are inherent ambiguities. In this work we use various cues and machine learning techniques in order to resolve these ambiguities. As just mentioned, instead of reprojecting every point in an image to its corresponding 3-D coordinate it is often more convenient to convert the image to some 3D representation (such as identifying major planes in the image), or even just infer 3-D information at each point such as the direction of the normal at that point. This allows the system to work with some assumptions which in turn assists in resolving the inherent ambiguities in the reprojection. In this work we look at such more convenient representations in both the indoor and outdoor context.

[0] *http* : //*izismile.com*/2010/03/17/*unbelievable_photos*_26_*pics.html*

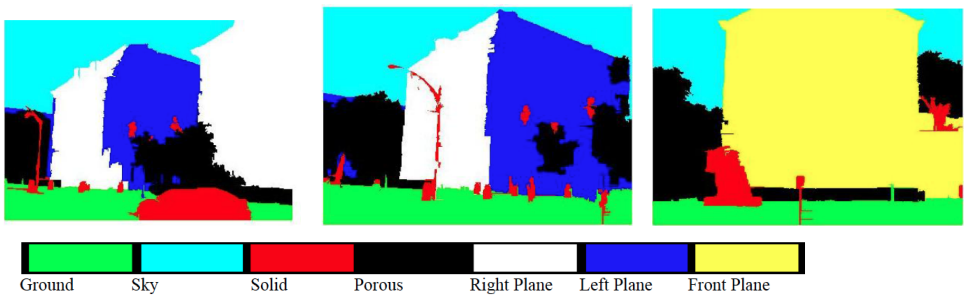


Figure 1.3: The top row shows images from the dataset overlaid with their ground truth labels, indicating which 3-D surface each pixel belongs to. An index for these labellings is beneath the images.

1.1 Overview of 3D Scene Reconstruction

In this section we will look at some examples of 3-D scene reconstruction and identify exactly the reprojections we hope to achieve in this paper. A more thorough overview of previous works is left till chapter 2.

1.1.1 Outdoor Scene Reconstruction

For outdoor scene reconstruction we work with a dataset we compiled consisting of stereo images that we collected in the Urbana-Champaign area. For each stereo pair we want to be able to get a 3-D representation of the scene (we focus on reconstructing the left image in the stereo pair).

In the outdoor context we aim to categorize each pixel into one of seven categories: ground, left facing plane, right facing plane, front facing plane, solid object, porous object and sky (shown in figure 1.3). This categorization was first presented in [4] and the authors also present a way of getting 3-D models given these categories using a "Pop-Up" method. We will go into further details in chapter 4.

1.1.2 Indoor Scene Reconstruction

For indoor scene reconstruction we follow the box layout assumption introduced in [5]. In this work the authors assume that room structure is in the shape of a box. Although this may seem to be an oversimplification, in the real world most rooms do follow this assumption. Even if they do not, by

Figure 1.4: The top row shows three images from the indoor scene dataset (details in chapter 3). The second row shows the groundtruth boxes which have been fit to these rooms.

fitting a box to the room we can get a fairly accurate estimate about its dimensions.
Examples showing sample images from the indoor image dataset and the corresponding ground truth boxes are shown in figure 1.4.

1.2 Summary of Approach

We have chosen to catergorize images into two broad categories: outdoor and indoor scenes. This is a natural categorization with few works attempting to handle both scenarios with the same technique. This is mainly due to the fact that different simplifying assumptions can be utilized in each case to remove the inherent ambiguities (e.g. in outdoor scenes foliage and sky pixels can be easily detected while in indoor scenes the entire scene can be assumed to exist in a cuboid box).

For the outdoor context we have chosen to integrate both single image and stereo cues. Stereo cues use known camera geometry and two slightly displaced views of a scene to come up with 3D information of the scene. On the other hand, single image cues are extracted from one image and use

knowledge from previously seen images to extract 3D information. We feel that an integration of both cues would be the best approach in this case.

For the indoor context we follow the general trend in indoor scene reconstruction [5, 6, 7] and focus on single image cues. We build on work presented by Hedau et al [5] and try to make the process more robust so that it can produce accurate results in more clutterred scenes.

1.3 Thesis Structure and Contributions

The following is an overview of the structure and contributions of this thesis:

Chapter 2. In this chapter we introduce some computer vision techniques that are relevant to the ideas presented in the rest of this thesis. We then mention current state of the art techiques for dealing with 3-D reconstructions in both the indoor and outdoor context.

Chapter 3. In this chapter we focus on the three datasets collected for the purpose of this work. We explain our reasoning for creating these new datasets (instead of just using existing ones) and describe the contents, collection and annotation process for each.

Chapter 4. In this chapter we focus on 3D reconstruction in the outdoor context. In this context we work with stereo images and focus on utilizing both single image and stereo cues in order to extract 3D information from the images. We also look at the calibration of stereo cameras since this is a necessary precursor to extracting stereo features.

Chapter 5. In this chapter we build on the machine learning techniques presented by Hedau et al [5] and look into ways these can be improved in order to improve the accuracies of our 3D reconstructions. We focus on structured Support Vector Machines and try to adjust standard formulations for our purpose.

Chapter 6. In this chapter we look back at all the methods presented in this thesis and focus on what we have learnt from each set of experiments. We also look ahead into further steps that can be taken and give a brief overview of how current techniques can be applied to much larger datasets.

CHAPTER 2

BACKGROUND AND RELATED WORK

Recovering 3D information from 2D images has long been one of the most active areas of research in Computer Vision. Work in this area can be divided into two categories: reconstruction from a single image and reconstruction from multiple images using stereo geometry. In this chapter we will give brief overviews of some computer vision concepts required to understand the techniques mentioned in this work. We will also go over the current state of the art in scene reconstruction for both the indoor and outdoor contexts.

2.1 Image Projections

The geometry of projections from the 3D world to a 2D image has been thouroughly studied, and using this camera model the effects of this projection can be explained. In this section we will give a brief overview of the pinhole camera model, which is a simple but insightful model of how modern cameras work. We will then use this model to explain the existence of vanishing points which play a key role in this work.

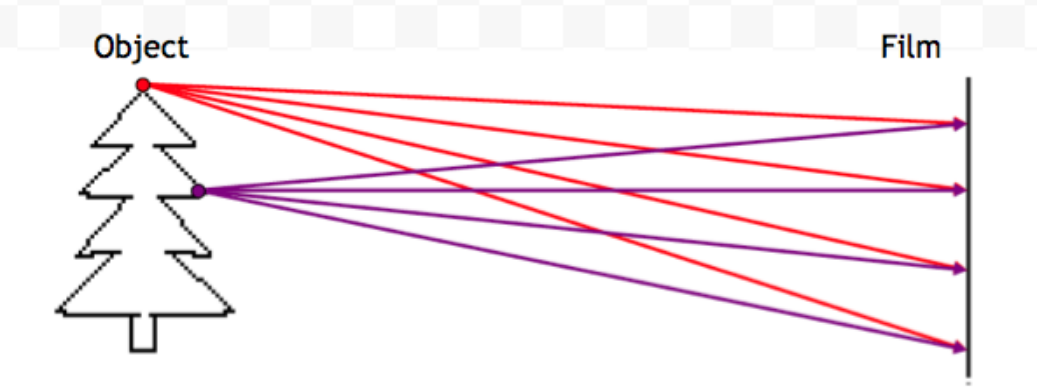


Figure 2.1: A simple camera setup in which a piece of film is placed in front of the object being photographed. Image from [8]

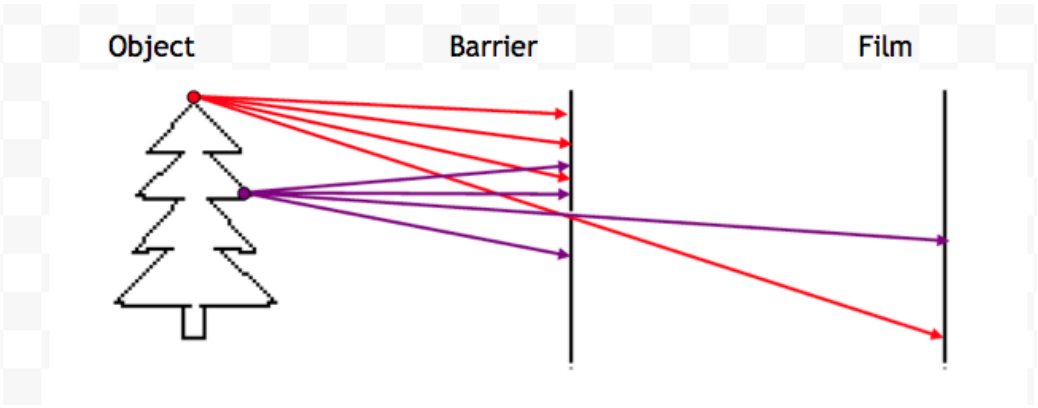


Figure 2.2: A pinhole camera setup in which light from the object being photographed reaches the film via a single hole. Image from [8]

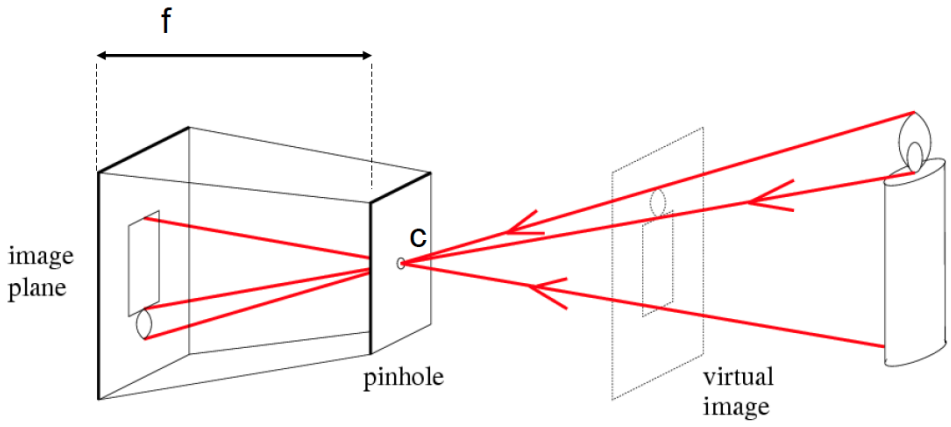


Figure 2.3: A detailed look at the pinhole camera model. c represents the camera center (the pinhole). The focal length is defined as the distance from the film to the camera center and is shown in the image as f. Image from[9]

2.1.1 Pinhole Camera Model

We see an object because light (originating from the sun or any other light source) is reflected off the object and enters our eyes. This light falls on receptors in our eyes, which send signals to our brains that are then deciphered as images. Since light bounces off objects in every direction theoreticaly it should be possible to create an image by just placing some film in front of an object. This setup is shown in figure 2.1.

However, as you can see that under this setup light rays from a single point will land on multiple points on the film. Indeed, for every point on the object and every point on the film, if there exists an unobstructed path between them there will be a transfer of light. Under these conditions the developed film will just show a blurry mess as multiple light sources hit the film at every point.

In order to allow only light from one point source to hit the film at a certain point, the pinhole is introduced. This setup is shown in figure 2.2.

In this model, light from an object only reaches the film via a single hole (the pinhole). If we can ensure that the pinhole is infinitesimally small we guarantee that for every point on the object there is only a single angle via which light can reach the film. This leads to a sharp image created on the film.

Although a modern camera is considerably more complex (due to lenses and the impossibility of an infinitesimally small pinhole), the pinhole model is a good approximation. A detailed version of the model is shown in figure 2.3.

In the next section this model is used to explain some of the phenomena we see while taking images.

2.1.2 Image Perspective and Vanishing Points

In figure 1.2 we introduced the concept of perspective distortion. In this image the toy car and the real car look indistinguishable even though we know that in the 3-D world the proportions of each car are very different. This is due to the relative positions of the objects with respect to the camera. Other examples of this phenomenon are shown in figure 2.7.

The reason behind this phenomenon is very easily explained by the pinhole camera model. If we look at figure 2.4 we can see that the angle an object

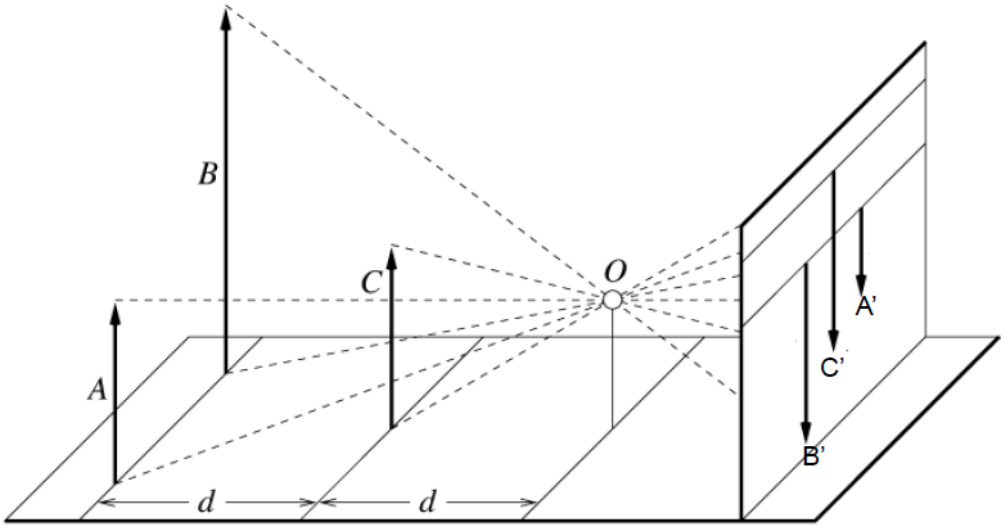


Figure 2.4: The pinhole camera model explaining the perspective distortion effect. Image from [9]

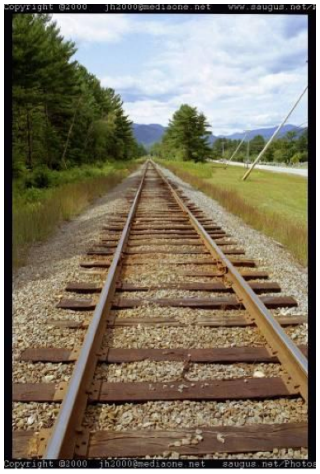


Figure 2.5: Perspective distortion causes train tracks (which we know are parallel) to appear to meet in the distance. Image taken from [11]

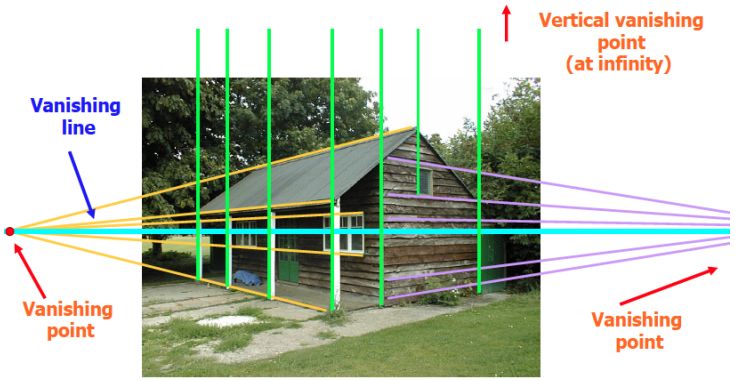


Figure 2.6: Buildings and other man-made structures mostly consist of lines lying in one of three orthogonal directions which lead to three major vanishing points. Image taken from [10]

Figure 2.7: Optical illusion caused by different positions of objects relative to the camera. Images taken from [1].

forms with pinhole determines its size in the image. In the figure you can see that although object A and C are of the same size, in the image object C appears to be much larger than object A. The is because object C subtends a much larger angle about the pinhole than object A. It is not difficult to see that the size of an object in an image is inversely proportional to its distance from the pinhole, thus objects that are further away seem to be smaller.

Vanishing points are a consequence of this distortion. Since distances that are farther away from the camera center appear smaller, lines that are parallel in the real world appear to come closer in images (unless they are parallel to the image plane). This can be seen when looking down train tracks, which appear to meet in the distance (figure 2.5). In man-made environments lines tend to fall in one of three mutually orthogonal directions which correspond

[1] http://www.synenergy-env.com, http://lustich.de/bilder/andere/schiefer-turm-von-pisa/ and http://presurfer.blogspot.com/2010/05/35-examples-of-forced-perspective.html

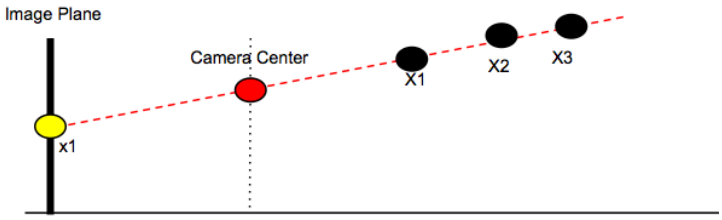


Figure 2.8: A point (x1) in an image is projected back into the world. The corresponding 3-D point can lie on any one of the points (X1,X2 or X3) along the projected ray

to the three coordinate axes. This is especially true in buildings and indoor scenes where structures are mainly rectangular and thus lines lie in either the X,Y or Z plane. Thus, in most such scenes we can find three major vanishing points which correspond to the meeting points of lines lying in the three orthogonal directions (see figure 2.6). Much use is made of these vanishing points, especially when trying to determine the 3D structure of buildings.

2.2 Stereo Geometry

The final computer vision concept we will overview is that of stereo geometry. While the subject is too large to analyze in detail we will provide an overview that should be sufficient for the reader to understand how stereo cues are used to enable outdoor scene reconstruction.

2.2.1 Geometry of Views

Given the point on the film and the position of the camera center (and some other camera parameters that we will not go into) it is possible to trace a ray from the film back into the world. The 3-D location that corresponds to this 2-D point may be located anywhere on this ray (figure 2.8).

However if we can observe the same 3D point in two camera views and know details of the relative positioning of the cameras we can reproject the point in both images and find the point of intersection of the two rays. Since this is a single point, we will be able to determine the 3D coordinate of the

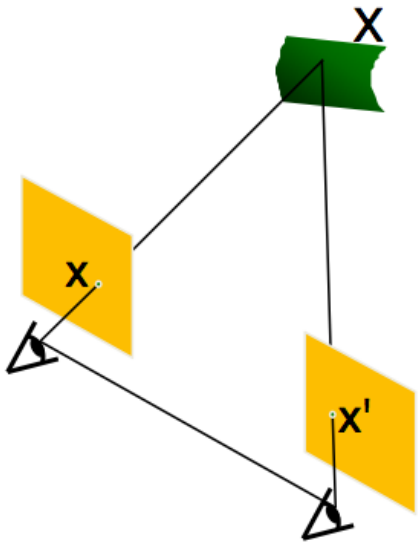


Figure 2.9: Given the 2D coordinate of a point in two different camera views and some knowledge of the two cameras, it is possible to find the exact 3D coordinate of the point. Image from [11].

2D point (see figure 2.9). It is this idea that lies at the heart of stereo reconstruction. It is also the relative positioning of objects in two different views that gives humans the ability to judge the 3D structure of objects, with the two views provided by our two eyes.

2.3 Related Work

Work on 3d reconstruction has traditionally focused on stereovision [12] which cannot robustly deal with all kinds of images. This is because stereo techniques depend on the ability to accurately detect corresponding points accross images which is not simple. Another approach utilizes structure from motion. However this requires multiple views of the same scene and again depends on points being accurately tracked across images. Attempts at reconstruction from a single image have also been made with ideas using constraints derived from shading [13] and known vanishing points [14].

In the outdoor context there there has been work that focuses on reconstruction from a single image [15]. This work was later extended to use both single image and stereo cues[16]. Hoiem et al. [4] worked under the as-

sumption that the ground is a single flat plane and used this assumption to "pop-up" objects and vertical planes to create impressive fly through videos of the scene.

In the indoor context, Hedau et al. [5] introduced the box assumption which simplified room structure to simple cuboids. This simplifying assumption allowed progress to be made with some research focusing on making the machinery more efficient [17] while others focused on incorporating other visual cues which could be used to make the box prediction more accurate [7]. Aproaches that move beyond this assumption have also come to the forefront recently such as work by Lee et al. [18].

CHAPTER 3

DATASETS

A key part of the work for this thesis was collecting new datasets for the problems tackled. We created three new image datasets which we will make available for anybody to use. In the following three sections each of the datasets is introduced and the rationale for creating them is provided.

3.1 Stereo Images of Outdoor Scenes

While focusing on outdoor scenes we decided to collect a dataset of stereo images of buildings. The stereo images were taken using the Fujifilm W3 3D camera and were of buildings in the Urbana -Champaign area. The images were then annotated manually, with polygons being marked and labelled as belonging to one of the following seven categories: ground, sky, left facing plane, right facing plane, porous object and solid object. The dataset consists of 67 such stereo pairs.

Examples of stereo image pairs are shown in figure 3.1. Examples of the left image in a stereo pair along with their corresponding ground truth labellings are shown in figure 3.2.

While there already exist many datasets of stereo images we felt the need to create a new one for the following reasons:

- Firstly, most current stereo pair datasets [19] focus on small scale alboratory scenes. We wanted a dataset in which the images captured street scenes, like the KITTI vision benchmark suite [20].

- Secondly, we felt there was a lack of stereo pairs with known camera parameters. By creating our own dataset we were able to ensure we had accurate camera calibrations for the two cameras.

- Thirdly we felt that datasets like [21] lack variables such as occluding or

Figure 3.1: Examples of stereo image pairs collected for our dataset. In each rom the image in the left column is the left image in the pair and the one on the right is the right image

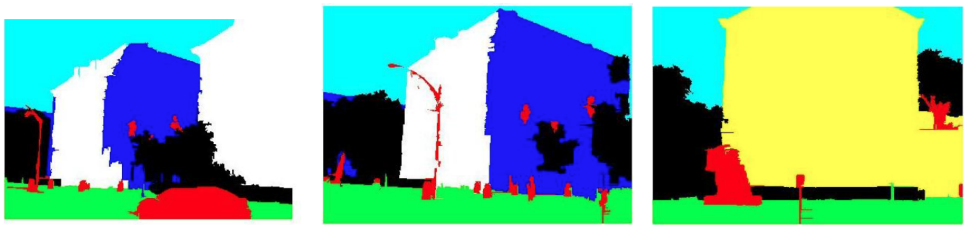


Figure 3.2: Examples of the left image in the stereo pairs shown overlaid with the ground truth annotations

moving objects which make the stereo calibrations and reconstructions more difficult. By creating our own dataset we were able to collect images that were more representative of images taken 'in the wild'.

3.2 Images with Groundtruth Vanishing Points

As mentioned in chapter 2, vanishing point detection pays a vital role in indoor scene reconstruction. Most current indoor techniques ([17, 18]) assume that accurate vanishing point detection techniques exist and move forward with that assumption. In our study of vanishing points we attempted to improve the detection methods. As a control we also ran reconstruction techniques using ground truth vanishing points to get their preformance using highly accurate vanishing point detections.

For this dataset we used the images in the dataset provided by Hedau et al in [5] which consisted of 314 images taken of indoor scenes. We then marked a few lines in each of the orthogonal directions in each image. The point that best fit the intersection of each group of lines was then used as the vanishing point. We used code provided by Lee et al. [18] in order to find the point of best fit.

Examples of images along with the lines marked in the orthogonal directions are shown in figure 3.3.

3.3 Indoor Images with Groundtruth Box-Layouts

The final dataset we collected was of indoor images labelled with groundtruth box layouts. Although a similar dataset has been provided by Hedau et al [5] and is extensively used by the vision community we felt that a new dataset was needed for the following reasons:

- Firstly, this dataset consists of only 314 images which we felt wasn't a large enough number to accurately evaluate the performance of our methods.
- Secondly, and most importantly, we feel that this dataset consists of very regular images. The rooms are mostly unclutterred and even if

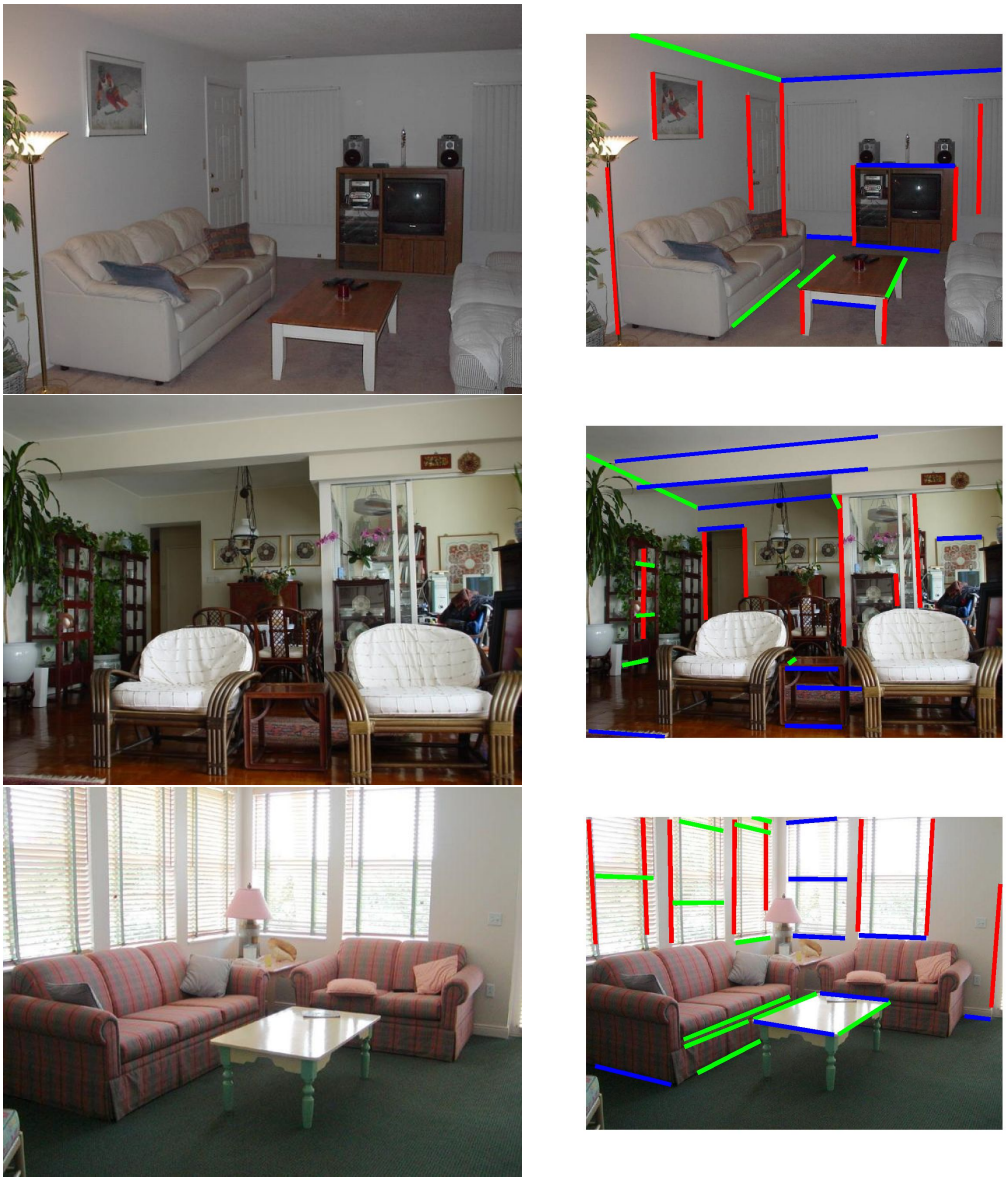


Figure 3.3: Examples of images shown along with the ground truth lines in each of the three orthogonal directions. These lines are then used to determine the vanishing points in each of these directions.

Figure 3.4: Examples of images in the new indoor dataset shown along with groundtruth annotations of the room box layout.

they are unclutterred the orientation of the objects tends to match the orientation of the room. Also, most of the rooms fit the box assumption very well which aids methods working with this assumption.

With this in mind we created a new dataset consisting of 650 images. These images were taken from the SUN dataset [22] and from the dataset provided by Hedau et al.[5]. The annotators (volunteers from the UIUC Computer Science Department) were asked to mark the box layout in each of the images. Some of the images marked are shown figure 3.4. As you can see, the new dataset consists of rooms with random clutter and rooms that do not follow the box assumption very accurately.

CHAPTER 4

OUTDOOR SCENES

In this section we talk about our approach to reconstructing outdoor scenes. We base our approach on the work of Hoiem et al [4] using bosted decision trees to classify groups of pixels into one of 7 different categories: Ground, Sky, Left Facing Plane, Right Facing Plane, Front Facing Plane, Porous Object and Solid Object. Given this categorization Hoiem et al provide a simple method for creating 3D models by assuming the ground is a flat plane that lies on the X-Z axis. Given this we can "pop up" objects and planes from the ground plane in a manner similar to that found in children's pop-up books.

4.1 Dataset

In this section we use the outdoor stereo image dataset introduced in chapter 2 which was specifically collected for this work. This dataset consosts of 67 pairs of stereo images taken in the Champaign-Urbana area.

4.2 Approach

For our work we follow the approach presented by Hoiem et al [4]. We start with an input image and take the left image as the one we are trying to reconstruct. We split the image into groups of similar pixels using the code provided by Pedro F. Felzenszwalb [23]. This reduces the number of data-points to be classified from about half a million pixels to a few hundred groups of pixels (or superpixels) for an average image. This greatly improves the performance of the system. Examples of images along with their superpixel groupings are shown in figure 4.1.

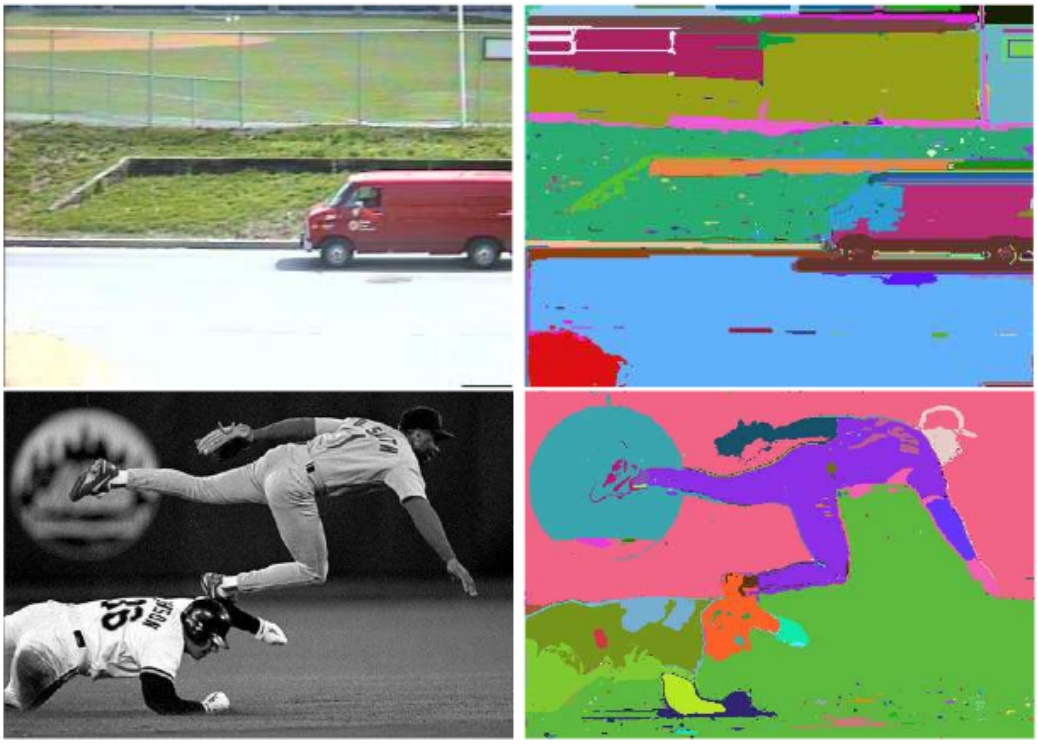


Figure 4.1: Examples of images collected for our stereo dataset. Each image shows the left image in the stereo pair. Image from [23]

Now we have these superpixels we extract features for each one. We have two sets of features: Single Image Cues and Stereo Cues. To extract single image features we use just the left imge. For stereo cues we use code provided by [24] to extract the 3D coordinates of each point in the left image. We then use these 3D points to extract some 3D features for each superpixel.

Given the groundtruth labellings of the superpixels and this feature set we train a classifier based using boosted decision trees, which we can then use to classify superpixels in the test images.

4.3 Single Image Cues

We focus on two types of features for our single image cues: color features and texture features. The details of the two are as follows:

Color Features Color is an important cue for identifying different materials. For example foliage is usually green while the sky is usually blue

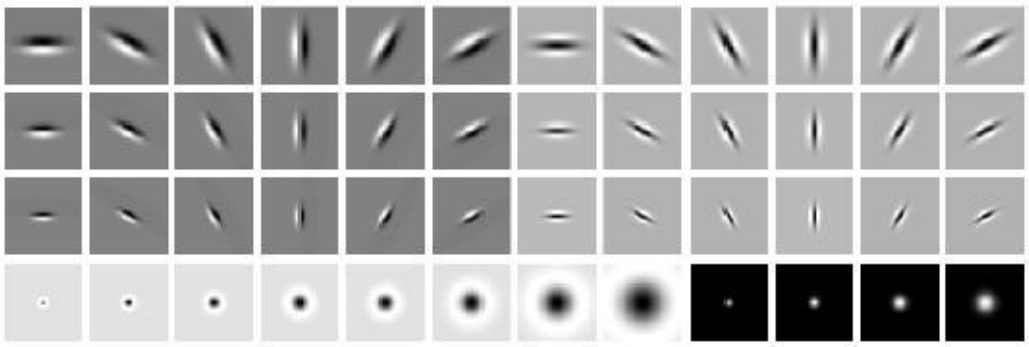


Figure 4.2: The LM filter bank has a mix of edge, bar and spot filters at multiple scales and orientations. It has a total of 48 filters - 2 Gaussian derivative filters at 6 orientations and 3 scales, 8 Laplacian of Gaussian filters and 4 Gaussian filters. [25]

or white. The color of a superpixel is represented by the mean RGB and HSV values of its constituent pixels.

Texture Features Texture cues are useful in distinguishing surfaces from each other. For example, buildings are usually composed of evenly spaced orthogonal lines while a clear sky is smooth. The texture of a superpixel is represented by the mean response at each of its component pixels to the Leung-Malik filter bank [25] . The LM set is a multi-scale, multi orientation filter bank with 48 filters. It consists of first and second derivatives of Gaussians at 6 orientations and 3 scales making a total of 36; 8 Laplacian of Gaussian (LOG) filters; and 4 Gaussians (see figure 4.2).

2D Coordinates The location of a pixel in an image is also a strong indicator of its category, e.g. the sky is usually present high in the image while the ground is near the bottom.

4.4 3D Features

The stereo images captured are used to create a stereo reconstruction of the scene using code provided by [24]. This gives us an estimate of the 3D location (factors such as small baseline relative to the scale of the scene and the instability of stereo algorithms in real world situations cause these stereo

estimates to be very unreliable) of each point in the scene, allowing us to extract the following 3D features:

Surface Normal For each Super-Pixel, a plane is fit to the 3D points corresponding to each pixel within it. From the plane equation

$ax + by + cz + d = 0$

the vector [a b c] is saved as the surface normal. This is a useful feature since buildings will usually have surface normal with a zero component in the y direction (assuming images are taken from ground level). The ground however will have an upward facing surface normal.

Depth Variance This is the variance of the z coordinate of the 3D coordinates corresponding to the pixels within the SuperPixel. For each superpixel the depth variance is defined as:

$$s_N = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

where x is the z coordinate of each point in the superpixel and n is the total number of such points.

This is used to differentiate front facing planes and objects such as cars which have little variation in depth from ground and left/right planes which have greater variation.

Point Variance from fitted plane This is the varaince in the Euclidean distance between the 3D points and the planes fitted to them. This plane should not makes sense if the object is non-planar, thus in objects such as trees the points should high variance while walls and the ground should show less variance.

A summary of the features used is given in table 4.1.

4.5 Predictive Model

For each stereo pair, the superpixels in the left image were passed through the feature extraction process. These, along with the ground truth labels, were passed to a boosted decision tree which creates models to classify each superpixel. Details of the model training can be found here [4].

Feature Description	Feature Size
Mean RGB	3
Mean HSV	3
Mean Respone to LM filters	48
Surface Normal	3
Depth Variance	1
Point Variance	1
2D Coordinate	2

Table 4.1: Summary of Features used

Figure 4.3: The image in the left column is reconstructed as a 3D model and novel views of the scene are shown in the images in the center and right.

4.6 Creating 3D Models

For each image, every super pixel is categories as either ground, vertical or sky according to its classification results. These are then input to the pop up algorithm described in [4] to create 3D models of the scene. An example 3D model is shown in figure 4.3.

	Porous	Solid	Ground	Left	Right	Front	Sky
Porous	.9936	.0176	.0714.	.0312	.0716	.0122	.0337
Solid	.1827	.8569	.3465	.1216	.3083	.0499	.0015
Ground	.0447	.0147	.9987	.0024	.0137	.0108	.0000
Left	.1305	.0448	.0348	.6002	.7184	.3210	.0195
Right	.1004	.0285	.0787	.2279	.9215	.2847	.0275
Front	.0675	.0338	.0753	.2955	.8048	.4951	.0923
Sky	.0117	.0002	.0001	.0025	.0184	.0017	.9998

Table 4.2: Confusion matrix showing how the superpixels are labeled. For each superpixel, the ground truth is considered the category into which the majority of its pixels are classified

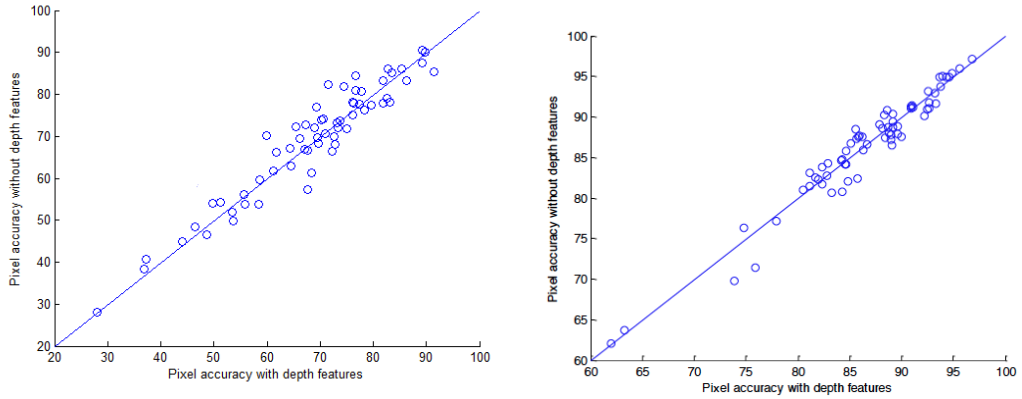


Figure 4.4: The left scatterplot shows the percentage accuracy of classification into 7 categories (sky, ground, solid, porous, front plane, left plane, right plane) with depth features (x axis) vs. without depth features(y axis). The right one shows the percentage accuracy of classification into 3 categories (ground, vertical, sky).

4.7 Results

Figure 4.4 shows a scatter plot of the pixel wise accuracies of the classifying the test images into the 7 general categories as well as the accuracies of classifying into broader regions (ground, vertical, sky). It is obvious from these plots that the stereo features are adding very little value to the classification in both scenarios.

A confusion matrix of the seven category scenario can be seen in table 4.2. This matrix confirms that the classifier is very accurate at labelling the objetcs, ground and sky catrgories. However when it comes to distinguishing individual planes the classifier is does not perform very well. Reasons that may account for this are discussed in the discussion section.
Figure 4.5 shows some sample results of the labelling system.

4.8 Clustering Vertical Regions

One of the advantages of using depth cues is the ability to cluster the vertical regions into individual planes instead of continuous segments. The advantage of doing this is that planes that are adjacent in an image and face the same direction are indistinguishable in the original formulation. This leads to inaccurate reconstruction as these will be clumped together as the same

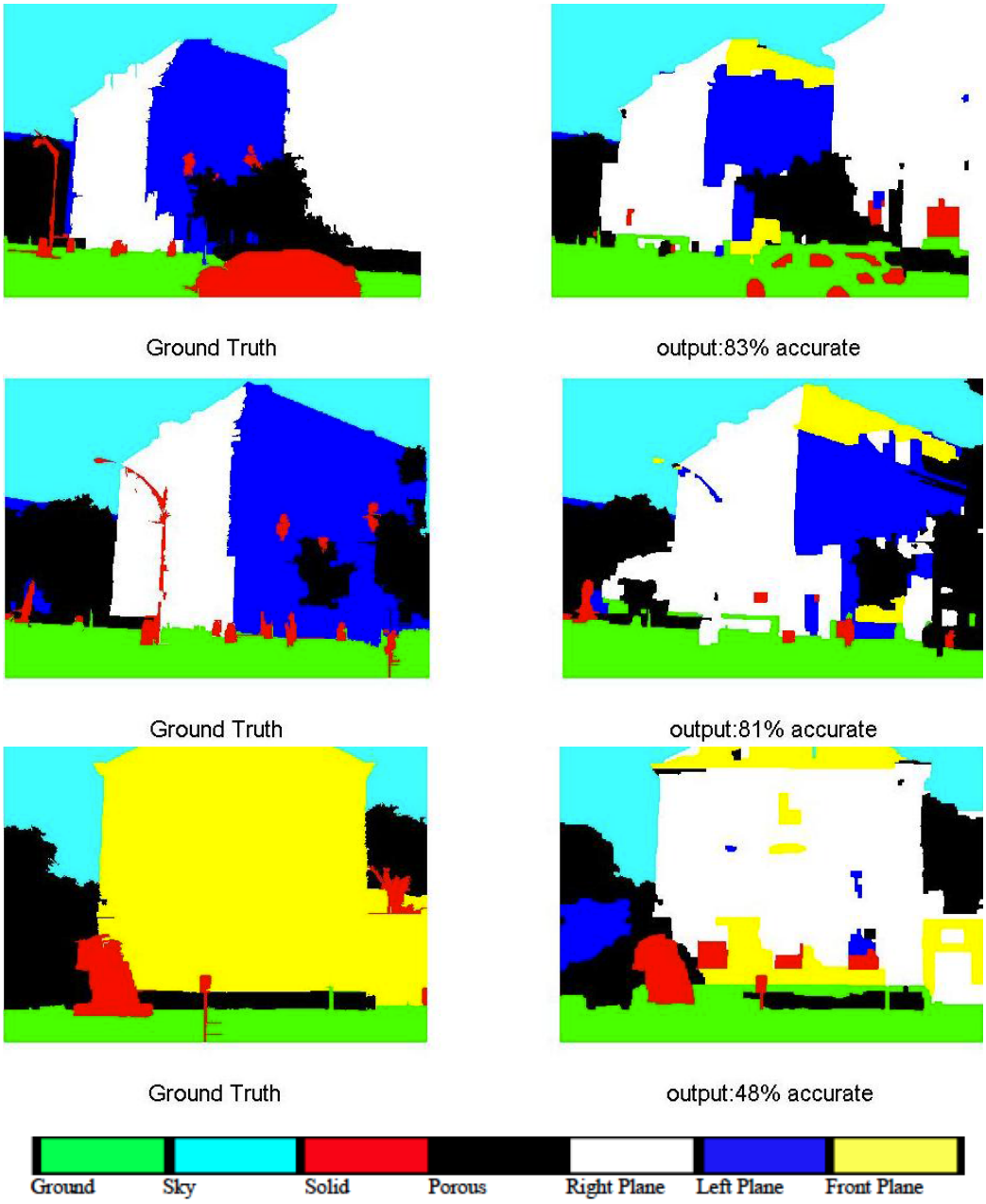


Figure 4.5: Sample results. The top two rows show the ground truth labels and the predicted labels for two images that have been labelled with high accuracy. The last row shows an image which has a low labelling accuracy due to the orientation of the plane being incorrect.

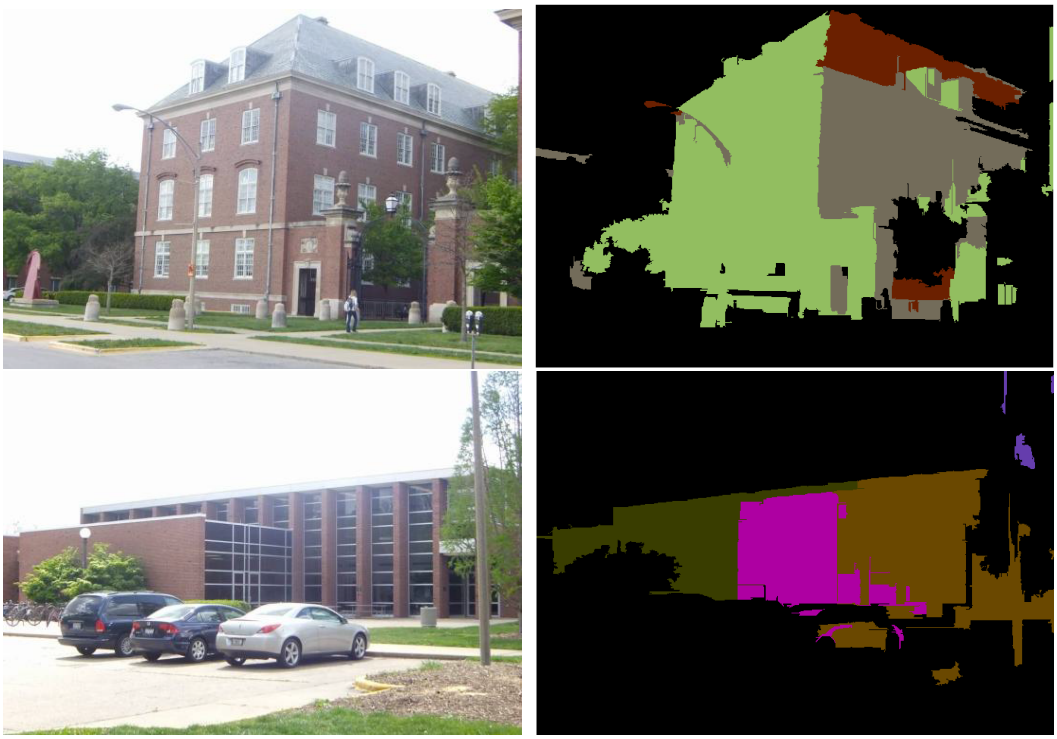


Figure 4.6: The left image shows the original image and the right images are these images with the vertical planes identified and clustered into individual planes

Figure 4.7: Plane clusters with the ground-plane boundaries shown in red

plane when they are "popped-up".

To do this the superpixels from the test images which have been classified as either a left facing plane, front facing plane or right facing plane are assembled and we extract a subset of the features used by the segment classifier. For each superpixel a feature vector is constructed which consists of the label assigned to it by the classifier, the surface normal of the segment and the mean coordinates of the pixels in the super pixels. The superpixel features are then clustered using the mean shift algorithm. The features in the feature vector are assigned weights so that more emphasis is placed on how the superpixels were classified, as this is usually an accurate way to distinguish between planes.

The results of this process are shown in figure 4.6. The plane classification is not perfect but you can see some adjacent planes with similar orientations that have been classified as separate planes. In figure 4.7 the boundaries between the ground and the solid objects have also been drawn to indicate where these objects would be popped up in the final 3D reconstruction.

4.9 Discussion

The main flaw in the approach described in this chapter lies in its failure to make full use of the depth information provided by the stereo images. Currently the 3D reconstruction requires the classification of pixels as either ground, vertical or sky. As the scatter plots in figure 4.4 show, a few simple single image features are able to do this to a reasonably high accuracy which makes the use of stereo features redundant. A better approach would be to classify the superpixels and then use the information about the location and orientation of the superpixels provided by the stereo system to do the actual reconstruction. This would allow for more complex scenes to be reconstructed.

But before this can be done it is obvious that the classification method must be improved. From Figure 4.4 it can be seen that the system classifies the superpixels into the three main regions reasonably well but has lower accuracy rates for the more specific labels. The confusion matrix in table 4.2 shows that this error lies mostly in the systems inability to distinguish between the left, right and front facing planes. A major cause of this error

lies in the ground truth, which has been marked by hand. In some cases a plane with a slight tilt away from the camera is marked as either left or right facing and in others a plane at a similar orientation is marked as front facing. There is also a bias against front facing planes, since the camera used to take the images had a narrow field of vision and thus it was often necessary to tilt the camera to get a decent view of the building from across the street. The solution to the mentioned problems may lie in the observation that as long as plane in the image is recognized as a plane and we have a fairly accurate depth map of the scene, there is no need to further sub classify planes. The vertical clustering described in section 4.8 further validates this claim. With this approach, the system would use single image features to classify superpixels as ground, sky or plane. The plane superpixels can then be clustered using coefficients of the plane fitted to each superpixel. The individual planes can then be accurately reconstructed.

Another flaw is the use of dense stereo correspondence. Every pixel is assumed to be accurately matched to its corresponding pixel in the second image. This is bound to be inaccurate. A better approach would be to score the correspondences depending on the corner-ness of the pixel or the similarity between the appearances of the correspondences and use only reliable pixels to fit a plane to each superpixel.

CHAPTER 5

INDOOR SCENES

In this chapter we focus on extracting 3D information from single images of indoor scenes. We work with both the dataset provided by Hedau et al in [5], as well as with the dataset we collected ourselves as described in chapter 3. Since our work builds on the work of Hedau et al we will introduce that work in the next section before going on to novel additions we made our this work.

5.1 Initial Approach

The details of the initial system can be found in [5]. Here we will give a brief overview of the method to motivate the following sections. An overview of the method can also be seen in figure 5.1.

1. The process takes an input image and extracts long lines from the image (figure 5.1 (A)). These lines are then grouped into three orthogonal directions using a slight variation of the RANSAC based approach introduced by Rother et al in [26].
2. The three groups of lines are then used to find three orthogonal vanishing points(figure 5.1 (B)).
3. The vanishing points are then used to generate candidate box layouts (figure 5.1 (C)). The details of this process are shown in figure 5.2. This is done by taking the vanishing points that are furthest in the horizontal and vertical directions and using them to generate layouts around the third vanishing point.
4. The top layouts are used to generate labels for each pixel (left wall, right wall, front wall, floor or ceiling). These along with other image

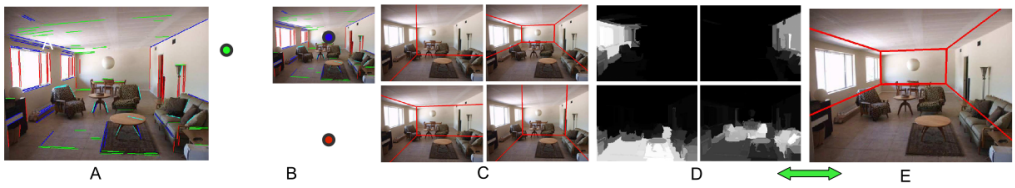


Figure 5.1: An overview of the method presented in [5]. The process takes a single image and perfrms edge detection to extract long line segments (A). These are then used to find three orthogonal vanishing points (B). The vanishing points are then used to generate possible box layouts which are then scored by the classifier to find the top scoring layouts (C). These initial layouts are used to find the surface labels (D, shwing labels for 'left wall','right wall','floor' and 'objects' respectively. The labels are then used to re-estimate the best scoring box layouts (E)). Images from [5].

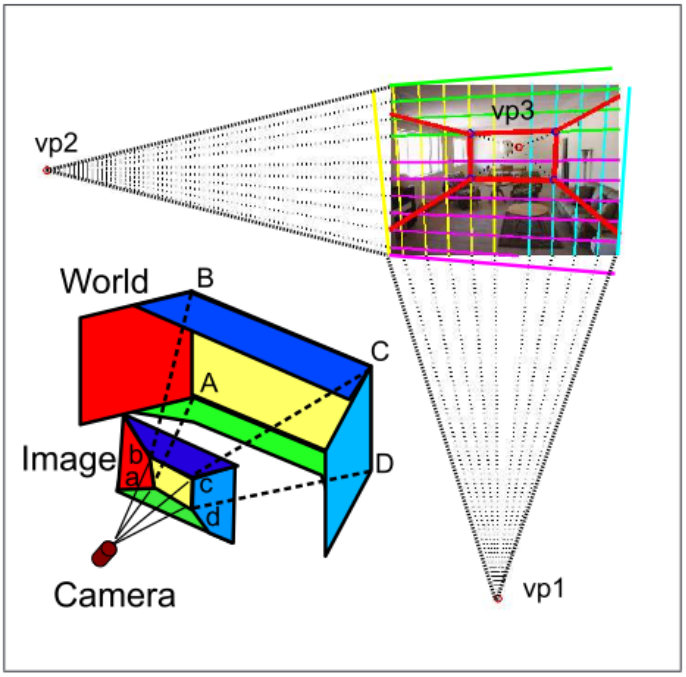


Figure 5.2: Using vp1 and vp2 (the vanishing points that occur furthest from the image center in the horizontal and vertical directions) as the vanishing points in the X and Y direction, candidate box layouts can be generated around the third vanishing point. Images from [5].

Experiment	Pixel Accuracy
Hedau et al	79.14
Self Similar Sketch lines	81.03

Table 5.1: Pixel accuracy using original method of Hedau et al and using self-similar sketch lines

features are used to generate surface labels for the pixels as described in [27]. The possible surface labels are 'left wall','middle wall','right wall','floor','ceiling' and 'object' (figure 5.1 (d))).

5. The surface labels are then used to generate new features which are then used to rescore the candidate box layouts leading to a new ranking. This process can be repeated untill the results are stable (figure 5.1 (E)).

5.2 Vanishing Point Detection

In the process mentioned above, as well as in other indoor scene reconstruction techniques (e.g. [18] and [7]) it is assumed that the vanishing point detection can be done accurately. However this is not always the case. The vanishing point detection may be inaccurate for multiple reasons including: (a) There aren't sufficient lines in each of the orthogonal directions, (b) clutter exists which is not aligned with the room orientation and misleads the vanishing point detector and (c) the lines in the orthogonal directions aren't detected due to occlusions. For all the three reasons presented, the presence of additional lines which agree with the room orientations would help improve the vanishing point estimates. With this in mind we experimented with the idea of SIFT lines to introduce additional lines into the vanishing point detection process.

5.2.1 SIFT Lines

SIFT features [29] have been standard image features used in computer vision research for a while now. Vedaldi et al intriduce the concept of self similar sketches [28], in which pixels are clustered together using their SIFT features.

Figure 5.3: (Left) Original images. (Center) SIFT descriptors used to cluster pixels into groups. (Right) Line fitted to individual groups to form self-similar sketches. Images taken from [28]

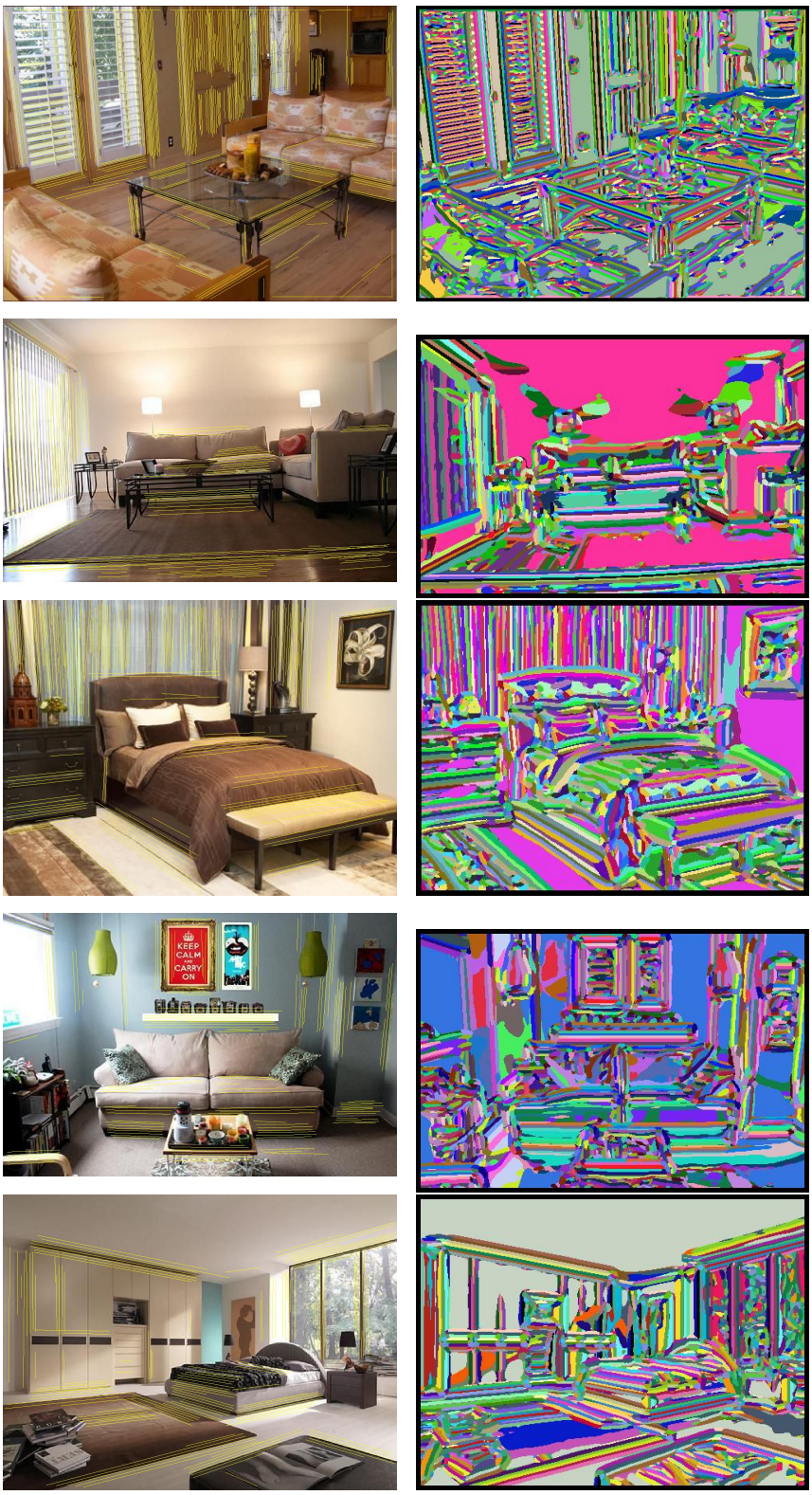


Figure 5.4: (Left) Indoor Scene images overlaid with self-similar lines. (Right) Pixels grouped by clustering on dense SIFT features

This leads to very similar pixels being grouped together and lines are fitted to these groups. In this way structures are detected which are not visible to standard edge detectors. Examples of these self-similar sketches can be seen in figure 5.3.

We use the lines provided by the self similar sketches in order to detect vanishing points by feeding these lines along with the lines detected initially to the original vanishing point detection method. We feel this is a good idea because as you can see in figure 5.3 (especially the third row which shows an indoor scene) the structures detected tend to follow the orientation of the room. Secondly, since RANSAC [30] is a majority based voting approach, we feel that the addional clutter lines should not hinder the performance of the vanishing point detection and the additional lines in the orthogonal directions can only help the detections. Finally, self-similar sketches give us a kind of structure that cannot be detected by edge detectors: contiguous empty spaces. For example in figure 5.3, in the third row you can see line structures are detected in the solid wooden planks of the bookshelf. Although this is not an edge, it is a kind of structure that aligns with the room and thus should be used in our vanishing point detection method.

Some examples of indoor scenes with the detected self-similar lines are shown in 5.4. The detected lines were then filtered so that only those longer than a certain threshold were kept.

The results of using these additional lines are shown in table 5.1. As you can see we are able to get a slight improvement using self similar sketch lines.

5.3 Support Vector Machines

The classifier trained by the above mentioned method is a structured support vector machine (Struct-SVM) [31] which uses ground truth box layouts and the extracted features from the training set to train the model to score candidate box layouts given an image. The Struct-SVM is formulated to train a scoring function which maximizes the score of structured labels that are close to the groundtruth labels. This "closeness" is decided by a user defined loss function: the smaller the loss between two structured labels the closer they are.

Hedau et al's original loss functions is composed of three basic losses which

function on corresponding faces of two given layouts. In these functions F_i refers to face i of the polygon. In the box layout assumption there are 5 faces. The three basic losses are as follows:

$$\delta_t(F_{ik}, F_k) = \begin{cases} 1 \text{ if } Area(F_{ik}) = 0 \text{ and } Area(F_k) > 0 \\ 1 \;\; Area(F_{ik}) > 0 \text{ and } Area(F_k) = 0 \\ 0 \text{ Otherwise} \end{cases}$$

$$\delta_c(F_{ik}, F_k) = \parallel c_{ik} - c_k \parallel^2$$

where c_i is the centroid of face i.

$$\delta_p(F_{ik}, F_k) = (1 - \frac{Area(F_{ik}) \cap Area(F_k)}{Area(F_{ik}) \cup Area(F_k)})$$

In the above formulations δ_t penalizes two layouts if a face is present in one and not in another. δ_c measures the distance between the centroids of corresponding faces and penalizes the two layouts proportionally. δ_p penalizes two layouts if the intersection of correponding faces is smaller than the union. This ensures that in high scoring pairs, the overlap between corresponding faces is high.

Using these individual losses the loss function is composed as follows. D_i represents the set of the losses δ_i over the five faces.

Hedau et al's Original Loss:

$$\triangle(y_i, y) = \sum_{k \in [1,5]} \delta_t(F_{ik}, F_k) + \delta_c(F_{ik}, F_k) + \delta_p(F_{ik}, F_k)$$

5.3.1 Loss Experiments

In this section we experiment with the loss definitions introduced in the previous section. The loss is designed to be a numeric measure of how different

Loss Function	Pixel Accuracy (%)
$\sum D_t + \sum D_c + \sum D_p$	78.85
$\sum D_t$	76.05
$\sum D_c$	79.22
$\sum D_p$	78.25
$mean(D_t)$	76.49
$mean(D_c)$	78.15
$mean(D_p)$	78.07
$max(D_t)$	75.85
$max(D_c)$	78.93
$max(D_p)$	77.95
$\sum D_t + \sum D_c$	76.76
$\sum D_t + \sum D_p$	79.18
$\sum D_c + \sum D_p$	78.57
$mean(D_t) + mean(D_c)$	77.84
$mean(D_t) + mean(D_p)$	78.96
$mean(D_c) + mean(D_p)$	78.03
$max(D_t) + mean(D_c)$	76.74
$max(D_t) + mean(D_p)$	77.96
$max(D_c) + mean(D_p)$	77.69
$.5 * \sum D_t + \sum D_c$	78.09
$.5 * \sum D_t + \sum D_p$	79.31
$.33 * \sum D_t + \sum D_c$	78.86
$.33 * \sum D_t + \sum D_p$	78.48

Table 5.2: Pixel accuracies of the loss experiments using the line detection method of [5]. In the table D_i refers to the set of loss δ_i as defined in section 5.3.

.

two candidate box layouts are. We experimented with the losses introduced in the previous section and used different weighted sums and statistical tools on the losses to see if we could induce an increase in accuracy. The original line detection formulation used in [5] was used for these experiments. The results are shown in table 5.2.

While the loss experiments may seem arbitrary, we were expecting certain experiments to perform better than others. We especially felt that by using only D_p or weighing it more than others we should get an improved classifier since the pixel accuracy, which is our evaluation metric, is most closely related to this loss. Pixel accuracy just measures the percentage of pixels in the predicted layout that have the same label as their groundtruth counterparts. It is obvious that if corresponding faces overlap, the pixel accuracy will increase.

However, as seen in table 5.2 changing the loss function had very little effect on the overall pixel accuracy.

Threshold	Number of Constraints	Training Time(s)	Accuracy(%)
No threshold	57717	6318.7	78.85
Less than 4	42277	2774.3	78.70
Less than 3.5	30198	1314.7	78.59
Less than 3	18385	356.99	78.41
Less than 2.5	9604	83.84	77.91
Less than 2	4278	13.93	77.45
Less than 1.5	1596	3.75	68.62
Less than 1	591	1.50	59.93

Table 5.3: The effect of limiting the number of incorrect layouts on training and testing. 'No threshold' represents the original SVM formulation in which all the incorrect layouts are used

.

5.3.2 Decomposed Learning

Another variation of the original Struct-SVM formulation comes in the form of decomposed learning. In the original formulation each training image had around 300 candidate layouts which were then compared with the groundtruth layout to train the classifier. If you view the Struct-SVM as just a linear clas-

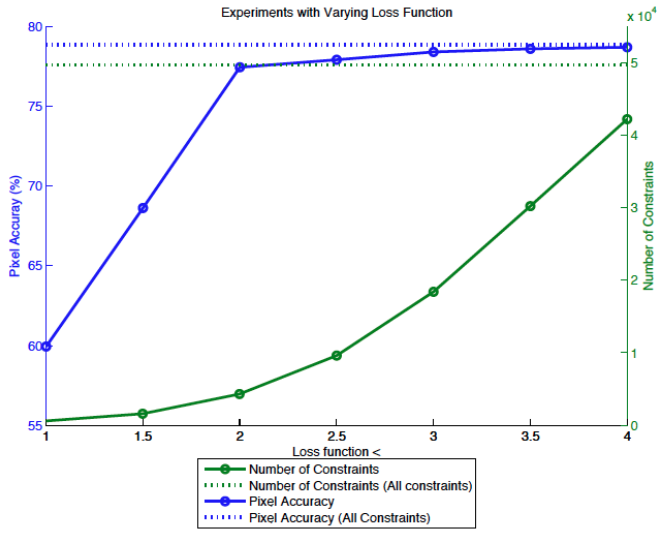


Figure 5.5: Graph showing the pixel accuracy (in blue) and the number of constraints in the SVM formulation (in green) as the loss function threshold is varied. As you can see the number of constraints can be greatly reduced before a significant drop in accuracy is observed

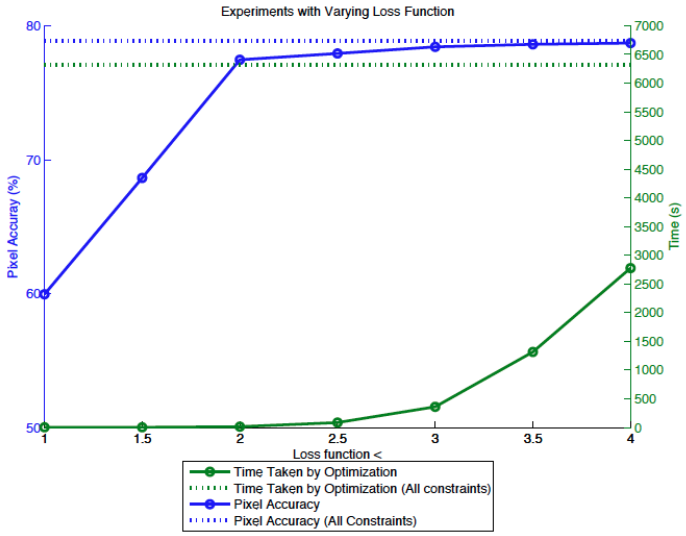


Figure 5.6: Graph showing the pixel accuracy (in blue) and the time taken for the SVM training (in green) as the loss function threshold is varied. As you can see the training time can be greatly reduced before a significant drop in accuracy is observed

sifier which divides an N-dimensional feature space using an N-dimensional hyperplane it is easy to see that candidate layouts which are far from the groundtruth are not as important to the classifier as those that are close by since if you are correctly classifying datapoints near your classification boundary you can be assured those far from the boundary are being correctly classified as well. This is the key idea behind decomposed learning introduced by Samdani et al in [32]. We apply it to our problem by thresholding the loss function values of our candidate box layouts during training.

We threshold the loss function so that candidate layouts with a loss greater than a certain value are ignored. The results are shown in table 5.3.The number of constraints show the number of candidate layouts that are compared with the ground truth layouts. As you can see this number can be decreased by a factor of ten without hurting the accuracy of the classifier.

The huge improvement in efficiency can be viewed graphically in figure 5.5 where we see that the number of constraints can be greatly reduced without seeing a significant drop in accuracy. A similar trend can be observed in figure 5.6 where the training time is significantly reduced before a significant drop in accuracy is observed.

5.4 Discussion

In this section we presented three novel additions to the original indoor scene reconstruction formulation provided in [5]: SIFT lines, additional loss functions and decomposed learning. The former two, although promising, did not significantly change the performance of the system. Decomposed learning however showed very promising results by reducing the time taken to train the SVM exponentially. This is significant not only from a debugging standpoint, allowing experiments to run faster over larger datasets, but could also lead to 'on the fly' training which could look for relevant training examples and train the classifier in real time.

CHAPTER 6

CONCLUSIONS AND FUTURE SCOPE

6.1 Conclusions

6.1.1 Outdoor Images

In chapter 3 we looked into integrating 3D and 2D cues to create 3D reconstructions from a pair of stereo images. The results show that if we follow the 7 category categorization used by [4] there is very little improvement of using both 3D and 2D cues over simply using 2D cues. However there are some promising insights to be gained from the work which can be summarized as follows:

- 3D cues can be useful in distinguishing between adjacent planes that face the same direction. This can be seen in the clustering algorithm presented in section 4.8 which is able to use a simple mean shift algorithm to distinguish between adjacent planes. Since single image cues would look the same for the two planes we can conclude that the 3D features are making this classification possible.

- The success of the clustering algorithm also points to the fact that the features used in the classification are powerful enough to distinguish between individual planes. This points to a reconstruction approach which focuses on the three category classification approach (sky, ground and vertical), with individual planes being reconstructed based on 3D features which can fit planes more accurately.

6.1.2 Indoor Images

In chapter 4 we focused on fitting a 3D box to a single image of a room. We tried three new techniques which gave us varying degrees of success. The main insights we learned from this line of work can be summarized as follows:

- Self-Similar lines provide useful additional information about structures in the scene which cannot be inferred from detected edges alone. While this additional information did not help with improving vanishing point estimations they could be utilized in other ways to understand indoor scene structure.
- Decomposed learning is a useful tool which can be employed in the structured SVM used in the formulation provided by Hedau et al [5]. It reduced training time exponentially and further steps should be taken to find ways to utilize this speed up to the advantage of room layout understanding.

6.2 Future Scope

6.2.1 Expanding the Dataset

One area in which we see huge potential for future work is that of expanding the ideas previously presented to run on large image datasets, on the order of millions of images. As a precursory step into this idea we explored the Houzz.com dataset (available at [1]). This dataset consists of approximately 1.4M images with each image belonging to one of the categories shown in table 6.1.

dining	kids	home-office	entry
deck	landscape	patio	porch
exterior	basement	garage and shed	hall
laundry room	staircase	wine cellar	

Table 6.1: Indoor room categories in the Houzz.com dataset

[1] http://tlberg.cs.unc.edu/memory/sirion/website/houzz.html

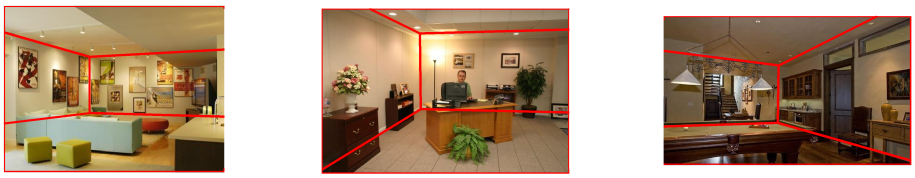


Figure 6.1: Sample images from the houzz dataset in which the method from [5] worked well. Results were evaluated qualitatively

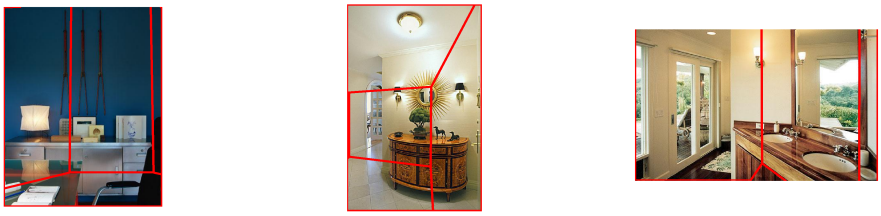


Figure 6.2: Sample images from the houzz dataset in which the method from [citation] failed due to inaccurate Vanishing Point detection..

Figure 6.3: Sample images from the houzz dataset in which the method from [citation] failed due to the image only focusing on objects

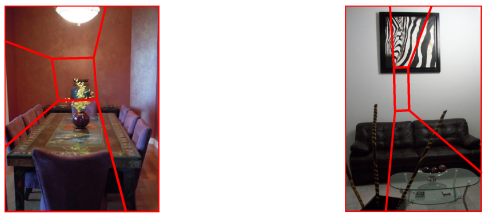


Figure 6.4: Sample images from the houzz dataset in which the method from [citation] failed due to physically implausible layouts being allowed in the predictions

We ran the spatial layout code provided by [5] on a subsample of the dataset just to see how the method would work on the dataset and to get an idea of challenges it would present. Since there were no ground truth labels for this data the results could only be analyzed qualitatively.

Figure 6.1 shows examples of images for which the method worked well. As is expected, the method performs well if the room captured in the image is box-shaped and is either relatively clutter free or the clutter is aligned with the orientation of the room.

However there are cases in which the method fails. These are as follows:

1. The method relies on the acurate detection of three orthogonal vanishing points. Vanishing point detection depends on the presence of sufficient lines in each of the three orthogonal directions defining the box that fits to the room. In the absence of sufficient lines the detection method may fail, causing the box layout predicted to be inaccurate. Images showing this error are show in figure 6.2.

2. The larger dataset also consists of images focusing on specific objects and not rooms in their entirety. The method, which expects to fit boxes using image cues about the room layout, fails in this scenario. Some images in which this error occurred are shown in figure 6.3.

3. Lastly the method has no priors that account for predicted layouts that are physically impossible. This leads to predicted boxes which are either impossible or highly unlikely to exist in real world scenarios. Some of these are shown in figure 6.4.

Thus there are specific challenges which a larger and less homogeneous dataset would present which need to be tackled.

6.2.2 Future Work

The main theme of this work has been to provide robustness to existing methods for reconstructing both indor and outdoor scenes. We feel that current methods depend too much on the input images being of a certain type. For example, in the indoor context rooms are expected to be box shaped with the majority of detected lines lying in one of three major orthogonal

directions. Thus these methods fail when presented with real world images with oddly shaped rooms or rooms full of random clutter. Similarly, if we use stereo techniques, unless we are able to perfectly match image points accross multiple images and ignore moving objects we will have incorrect reconstructions. The previous section showed that this problem is exacerbated in larger datasets. Since Computer Vision, and Machine Learning in general, is moving in the direction of using huge datasets, robust techniques are required that can handle the variety in input images.

We explored some techniques to increase robustness and feel that further steps can be taken to explore this challenging problem. We feel most enthusiastic about utilizing massive image datasets for this purpose. The Houzz.com dataset, for example, has great potential to improve the indoor reconstruction problem. Since the number of possible room layouts is limited, given a large enough dataset we feel it should be possible to match test images to images in the dataset and find close matches from which to transfer the scene layout. This would also remove the problem of implausible layouts generated by current methods. Such methods could be used in conjunction with current techniques to create more accurate reconstruction systems.

REFERENCES

[1] I. Endres, V. Srikumar, M.-W. Chang, and D. Hoiem, "Learning shared body plans," in *CVPR*, 2012.

[2] P. Dollr, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *CVPR*, 2009.

[3] M.Andriluka, S.Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *CVPR*, vol. 1, 2009.

[4] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic photo pop-up," in *ACM SIGGRAPH*, 2005.

[5] V. Hedau, D. Hoiem, and D. Forsyth, "Recovering the spatial layout of cluttered rooms," in *ICCV*, 2009.

[6] A. Schwing and R. Urtasun, "Efficient exact inference for 3d indoor scene understanding," in *ECCV*, 2012.

[7] S. Ramalingam, J. Pillai, A. Jain, and Y. Taguchi, "Manhattan junction catalogue for spatial reasoning of indoor scenes," in *CVPR*, 2013.

[8] S. Seitz, "Cameras [lecture presentation].retrieved from http://courses.cs.washington.edu/courses/cse455/12wi/lectures/projection.pdf," 2012.

[9] D. Forsyth and J. Ponce, *Computer Vision: A modern Approach*, 2nd ed. New Jersey: Pearson, 1964.

[10] A. Criminisi and R. Thomas, "Getting into the picture," Jan. 2003. [Online]. Available: http://plus.maths.org/content/getting-picture

[11] D. Hoiem, "Epipolar geometry and stereo vision [lecture presentation].retrieved from http://www.cs.uiuc.edu/ dhoiem/," 2012.

[12] D. Scharstein and R. Szeliski., "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, p. 47, 2002.

[13] R. Zhang, P. Tsai, J. Cryer, and M. Shah, "Shape from shading: A survey," *IEEE PAMI*, p. 690, 1999.

[14] A. Criminisi, I. Reid, and A. Zisserman., "Single view metrology," *IJCV*, pp. 123–148, 200.

[15] A. Saxena, H. Chung, and A. Ng, "Learning depth from single monocular images," in *NIPS*, 2005.

[16] A. Saxena, J.Schulte, and A. Ng, "Depth estimation using monocular and stereo cues," in *IJCAI*, 2007.

[17] A. G. Schwing and R. Urtasun, "Efficient exact inference for 3d indoor scene understanding," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 299–313.

[18] D. C. Lee, A. Gupta, M. Hebert, and T. Kanade, "Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces." in *NIPS*, vol. 1, no. 2. Vancouver, BC, 2010, p. 3.

[19] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[20] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[21] T. Werner and A. Zisserman, "New techniques for automated architectural reconstruction from photographs," in *Computer VisionECCV 2002*. Springer, 2002, pp. 541–555.

[22] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*. IEEE, 2010, pp. 3485–3492.

[23] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[24] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[25] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.

[26] C. Rother, "A new approach to vanishing point detection in architectural environments," *Image and Vision Computing*, vol. 20, no. 9, pp. 647–655, 2002.

[27] D. Hoiem, A. A. Efros, and M. Hebert, "Putting objects in perspective," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 3–15, 2008.

[28] A. Vedaldi and A. Zisserman, "Self-similar sketch," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 87–100.

[29] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[30] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[31] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 104.

[32] R. Samdani and D. Roth, "Efficient decomposed learning for structured prediction," *arXiv preprint arXiv:1206.4630*, 2012.