Improving Object Detection using 3D Spatial Relationships

by

Tristram Southey

MSc., University of British Columbia, 2006

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Computer Science)

The University Of British Columbia
(Vancouver)

August 2013

Abstract

Reliable object detection is one of the most significant hurdles that must be overcome to develop useful household robots. Overall, the goal of this work is to demonstrate how effective 3D qualitative spatial relationships can be for improving object detection. We show that robots can utilize 3D qualitative spatial relationships to improve object detection by differentiating between true and false positive detections.

The main body of the thesis focuses on an approach for improving object detection rates that identifies the most likely subset of 3D detections using seven types of 3D relationships and adjusts detection confidence scores to improve the average precision. These seven 3D qualitative spatial relationships are adapted from 2D qualitative spatial reasoning techniques. We learn a model for identifying the most likely subset using a structured support vector machine [Tsochantaridis et al., 2004] from examples of 3D layouts of objects in offices and kitchens. We produce 3D detections from 2D detections using a fiducial marker and images of a scene and show our model is successful at significantly improving overall detection rates on real world scenes of both offices and kitchens.

After the real world results, we test our method on synthetic detections where the properties of the 3D detections are controlled. Our approach improves on the model it was based upon, that of [Desai et al., 2009], by utilizing a branch and bound tree search to improve both training and inference. Our model relies on sufficient true positive detections in the training data or good localization of the true positive detections. Finally, we analyze the cumulative benefits of the spatial relationships and determine that the most effective spatial relationships depend on both the scene type and localization accuracy. We demonstrate that there is no one relationship that is sufficient on its own or always outperforms others and that a mixture of relationships is always useful.

Preface

This dissertation is an original intellectual product of the author Tristram Southey. All of the work presented henceforth was conducted in the Laboratory for Computational Intelligence.

A version of Chapter 4 appeared in [Viswanathan et al., 2011]. All stages of this work were performed jointly with Pooja Viswanathan and it is included with her permission. J. J. Little and A. Mackwork were involved in project developments in a supervisory position.

A version of Chapters 7 and 8 appeared in [Southey and Little, 2012]. I was lead research for all all material in this paper. J. J. Little was involved in project developments in a supervisory position. David Meger provided some software for assistance with the scene structure labelling component.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

A better understanding of the structure of human environments and the ability to make predictions about object type based on that structure could be useful for many applications. With the increasing availability of crowd sourced object labeling services like Mechanical Turk and 3D shape measuring devices like the Kinect stereoscopic camera, it is likely that we will soon have access to large quantities of accurate quantitative data about the spatial structure and makeup of human environments. At the same time, there is an increasing demand for home-based robots that can assist people with basic, everyday problems like fetching or putting away objects, cleaning houses and reminding people where they left their keys. All of these tasks require some level of knowledge about the structure of human environments.

Overall, the main goal of our work is to demonstrate that it is possible to significantly improve the accuracy of object detection using 3D qualitative spatial relationships. Our approach is based on the fact that the spatial relationships between objects in organized human environments exhibit structure. This predictable structure can be used to improve object detection rates by identifying a set of detections that have a layout in 3D that corresponds to a model of object relationships in that type of scene. For example, in a kitchen you are more likely to find a frying pan on a stove than on a refrigerator. Therefore, objects detected on stoves are more likely to be frying pans than objects detected on top of refrigerators.

Manually identifying all the rules that work together to determine object layout in houses is impractical. The scale of the problem is daunting, given the variety of objects and the multiple locations they can reasonably appear. Humans would have great difficulty trying to identify all the rules that govern object layout, let alone the probabilities behind these rules that allow us to classify an object based on its context. Learning provides us with an approach that can identify these rules with minimal human intervention and which can be easily updated for new objects and environments. To simplify the learning, we propose using a supervised learning approach that trains from labeled training data.

An unsupervised approach that could work with unlabeled data would require less data preprocessing would be preferable but it is not unreasonable to require object labels given the difficulty of the problem and the increasing prevalence of labeled data sources such as LabelMe [Russell et al., 2008].

What is needed is a mechanism that would allow a robot to take examples of object layouts in houses and develop a model of the commonalities between these layouts. This model would then allow the robot to make predictions about the type and layout of objects based on input from an object detector. Significantly, the locations of objects in houses are based only partially on the shape of the house. Instead, object locations are predicated on each other. A room is a bedroom primarily because of its contents and the location of an object in that room will be dependent on the location of other objects.

1.1 Spatial Knowledge in Action

To better understand our goals, let us examine an anecdotal example of how a human might apply spatial knowledge to a classification task. A person is sitting on a chair at a table and in front of them are three objects in a row. The object on one side is a knife and the object on the other is a fork. However, the object in the middle is covered and unknown. Given the scene, what type of object is in the middle? Most humans in this situation would be able to easily determine that it is most likely a plate.

This may seem like a trivial example but this classification is quite impressive since a plate would be a difficult object for many visual object classifiers to identify. Plates can vary in color and size, have few distinguishable features in their appearance and look similar to many other circular objects found in human environments. However, a person can identify the unknown object as a plate using only its qualitative position relative to the other surrounding objects because they have learned about the spatial relationships that exist between objects in human environments and how to apply those relationships to the problem of object classification.

1.2 Why Qualitative 3D Spatial Relationships?

The main difference between our work on using context for improving object detection and other approaches is the use of qualitative 3D spatial relationships to model the spatial layout of objects. Qualitative spatial relationships, such as *near*, *far* and *above*, are terms

humans use to conceptualize and communicate about their environment. The assumption behind the use of qualitative spatial relationships in human environments is that by basing relationships on human terms used to describe and discuss their surroundings, then the relationships contain the relevant information used by the humans when creating that layout. Qualitative spatial relationships are applied to decompose and partition the qualitative spatial relationships between objects that are produced by a robot's sensors, breaking the relationships down into multiple simplified representations based on some property of space such as topology, orientation or distance. Qualitative spatial relationships are used on problems with a spatial element because they simplify learning by translating complex spatial interactions into discrete values, reducing the complexity of the problem.

Most contextual object detection models that use spatial relationships are limited to relationships between detections in a 2D image space. Since 2D relationships are viewpoint-dependent, relations between the same objects can be inconsistent from image to image. Furthermore, 2D spatial relationships are limited in their expressiveness, unable to capture some spatial relationships that are clear in 3D. For example, it is very difficult to reliably determine if objects are on the same surface or directly above each other from a single image. 3D spatial relationships are viewpoint-independent and can provide richer spatial descriptions on which to base a model. 3D spatial relationships between objects are more consistent across different scenes than 2D spatial relationships as they are not view point dependent. Furthermore, they are more descriptive because they can capture a broader range of object arrangements. Therefore, we anticipate that 3D spatial relationships, if accurately localized, can provide a better basis for performing margin reweighing than 2D relationships.

Historically, 3D relationships have not been used due to a lack of data on the 3D layout of indoor scenes. Computing the 3D layout of objects in scenes is time consuming and difficult. In our work we have explored using both synthetic data sets from video games and produced our own data set of 3D scenes because there was no existing data set of 3D real world scenes to learn from.

1.3 Improving Object Detection Using 3D Qualitative Spatial

The following is a set of tasks that could be performed using a model relative object positions in houses:

1.3.1 Scene classification Using Object Detections

This is the problem of using object detections to determine the type of a scene. Scenes are either rooms or portions of a room devoted to a task which have a commonly used semantic label (e.g., office, kitchen, bedroom, bathroom, etc). Since scenes are task related, they rely on the presence of commonly used objects and many scenes are primarily just a collection of objects in an area. The difference between a bedroom and a office is mostly the contents, not the properties of the room itself. Determining the type of a scene based on raw object detection was important to our work as it allowed us to use scene dependent spatial models. We demonstrate that visual object detections could be used to recognize several common types of scenes with sufficient accuracy that we could use scene dependent spatial models.

1.3.2 Single Object Classification Through Spatial Context

This is the problem of classifying an individual object based on perfect information about the type and location of all surrounding objects. This early work was intended to test the feasibility of using qualitative spatial relationships to classify objects. We used synthetic data from a video game which contained many realistically structured houses to provide the training and test data. Our results demonstrate that it was possible to infer the type of many objects using only information about the qualitative relationships with their surroundings.

1.3.3 Improved Multi-object Detection Using 3D Spatial Relationships

The most significant area of this thesis is devoted to the problem of taking the results of a visual object detector, applied to multiple views of a scene and using a model of the spatial relationship between objects to reject false positive detections.

Our method identifies the most likely subset of 3D detections using seven types of 3D spatial relationships and adjusts detection confidence scores to improve the average precision. A model is learned using a structured support vector machine (SVM) [Tsochantaridis et al., 2004] from examples of 3D layouts of objects in offices and kitchens. This approach of using a structured SVM to improve object detection accuracy was applied to images in [Desai et al., 2009].

We describe a technique for generating 3D detections from 2D image-based object detections and demonstrate how our method improves the average precision of these 3D detections. We show that our approach could improve on the detection rates in real world scenes. After that we tested our method on synthetic detections to determine how factors such as

localization accuracy, number of detections and detection scores change the effectiveness of 3D spatial relationships for improving object detection rates. We end with an analysis of the relative performance of our seven types of spatial relationships.

1.4 Contributions

The following are the main contributions of this thesis in the order they are presented:

Scene Classification using Object Detections: We present a novel technique for performing scene classification using the results of an image-based object detector. We discuss the problems associated with determining the correct types of object to use for classifying scenes and our approach which uses alternating boosted decision trees.

Qualitative Spatial Relationships in 3D: We present a detailed analysis of qualitative spatial relationships and the the motivations behind their use. We describe the major types of qualitative spatial relationships commonly used and the techniques used to perform the measurements and partitioning necessary to decompose quantitative spatial relationships into qualitative ones. We focus on the problems of translating qualitative spatial techniques from 2D spaces where they are normally used into 3D. We then describe in detail seven qualitative spatial relationships that we believe provide a broad coverage of qualitative spatial techniques that are appropriate to our problem.

3D Qualitative Spatial Relationships and Single Object Classification: We present an analysis of the potential for 3D qualitative spatial relationships to be used to perform classification. This approach uses Alternating Boosted Decision trees as the learning mechanism and relies on synthetic training and test data from the video game Elder Scrolls 3. We demonstrate that classification with moderate accuracy is feasible using only information about surrounding objects. This success shows that spatial relationships might be very useful for improving object detection rates by removing false positive detections. This section also includes an analysis of the use of synthetic data for spatial problems such as these, with a description of the benefits, problems and essential elements required when using a video game for this kind of problem.

3D Qualitative Spatial Relationships for Improving Multi-object Detection: We present an approach for taking multiple images of a scenes, producing 3D object detections, identifying a subset of detections that comprise a likely scene according to a

learned model and then adjusting the detection scores of all 3D detections to remove false positive detections. We demonstrate our approach on real world scenes from IKEA and our university department.

The technique we used is based on that described in [Desai et al., 2009] for improving detection rates in 2D but differs and improves on their work in several ways. Firstly, we produce 3D detections, use 3D spatial relationships and use 3D training data. Secondly, we use a superior inference technique using a branch and bound tree search to identify better true positive subsets, this technique improves on the learning stage as well. Finally, we perform a much broader analysis of the properties of the object detections which lead to significant improvements from the use of this approach. We used synthetically generated scenes based on real world data to examine the role of detection localization accuracy, number of detections per scene, scene type and the average difference in score between true and false positive detections. Finally, we analyze the seven 3D qualitative spatial relationships to see how they perform relative to each other and discuss which relationships are effective under different circumstances.

1.5 Collaborations

During work on this thesis I collaborated on research with two individuals. The work described in Chapter 4 on Scene Classification using Object Detection was performed collaboratively with Pooja Viswanathan. We both contributed to the research, coding, experimentation and paper writing. In Chapter 8 on Object Detection using 3D Spatial Relationships I extended a code base created by David Meger for labeling objects in scenes in 3D.

1.6 Chapter Overview

The following is a list of the remaining of the chapters in this thesis and a brief description of their role:

Chapter 2: Related Work This chapter provides an overview of two learning techniques used in later chapters: Alternating Boosted Decision trees and Support Vector Machines.

Chapter 3: Context and Object Detection This chapter begins with an overview of

the core concepts behind object detection and a Deformable Parts model [Felzenszwalb et al., 2008], the main visual object detection technique used in this thesis. We discuss the many ways that context is used for learning problems, the ways it can influence a scene and the major sources of contextual information. The chapter ends with a broad examination of the approaches for applying context to multi-object and 3D object detection problems.

Chapter 4: Scene Classification using Object Detections: This chapter covers our approach to performing scene classification using the results of a visual object detector.

Chapter 5: Qualitative Spatial Relationships This chapter provides an overview of qualitative spatial relationships, the motivation behind their use and the core concepts related to them. We cover the four major types of qualitative spatial relationships, major approaches to measurement and partitioning and how to translate them into 3D.

Chapter 6: Spatial Object Classification in Virtual Environments: This chapter covers our early work on performing classification of a single object given perfect information about its surroundings. This chapter also covers the advantages and problems involved in using synthetic data from video games for learning object spatial relationships.

Chapter 7: Improving Object Detection using 3D Spatial Relationships: This chapter describes our approach to improving object detection accuracy by removing false positive detections using a learned model of expected 3D spatial relationships between objects in indoor scenes. This chapter also covers the seven qualitative spatial relationships we use with this approach.

Chapter 8: Object Detection using 3D Spatial Relationships Results: This chapter describes the experimental results of the model from the previous chapters applied to real world scenes. We describe our approach for producing 3D detections from 2D detections in multiple images and our data collection and annotation techniques for producing our 3D training and test data. We give results of our model applied to object detections computed on these real world scenes. We then describe our approach for producing synthetic detections with controllable statistical properties from these real world scenes. We describe experiments which examine how the properties of the hypothesis boxes used for training and test affect the final improvement from our techniques. We end with an analysis of the individual benefits from each of the seven qualitative relationships used.

Chapter 9: Conclusion and Future Direction: This chapter gives an overview of the conclusions we reached from this thesis. It ends by describing future directions for this research.

Chapter 2

Related Work

This chapter contains related research relevant to this thesis with a focus on the learning techniques we will use. In the next chapter we will examine the role of context in object detection and research in that area.

The goal of the first section of this chapter is to acquaint the reader with the principles and techniques involved in applying Boosted Alternating Decision Trees which we used in our early work in Chapter 6. Next we examine Support Vector Machines (SVMs) and provide a closer an examination of Structured Support Vector Machines [Tsochantaridis et al., 2004], sufficient to explain how they are used with our model for improving 3D detection rates in Chapter 7.

2.1 Boosted Alternating Decision Tree

Decision trees are a hierarchical model for supervised learning that identifies local regions in the input space using a sequence of recursive rule-based divisions of the data [Quinlan, 1986]. The general structure of decision trees has internal decision nodes that implement test functions with discrete results and terminal leaves that identify the classifier output. Classification generally works by starting at the root and recursively applying decision node test functions until a leaf node is reached. Most decision tree learning approaches are greedy, finding an optimal split according to an *impurity measure*. A split is "more pure" if the rule on which the data is split divides the data into two sets containing samples that belong to the same class, with splits containing a similar number of samples in each set preferred over unbalanced splits.

2.1.1 Adaboost and Boosted Decision Trees

Boosted decision trees [Dietterich, 2000] are a class of decision tree learning algorithms that apply Adaboost learning techniques to the iterative construction of decision trees. Boosting is a learning technique that determines how to combine and weight many weak classifiers to produce a single strong classifier. Adaboost (adaptive boosting) by [Freund and Schapire, 1999] was an improvement on earlier boosting techniques that built classifiers iteratively and adapted them based on the performance of previous iterations. The Adaboost algorithm focuses on so called "hard to solve" samples from the training distribution by increasing weights on samples incorrectly labeled by previous iterations and decreasing weights on those that were successfully classified. As a learning approach it is distinct from the underlying model it is constructing and so can be applied to many different type of models. In our work, we concentrate on boosted decision trees. Adaboost has seen application in a variety of research topics such as face recognition [Viola and Jones, 2004] and text classification [Schapire and Singer, 2000].

Boosted decision trees have several useful properties that make them appropriate for our work:

- They provide a classification score and therefore allow for a reject option.
- They require no knowledge about the properties of the weak classifiers used and can be combined with any weak classifier that is more accurate than a random guess, allowing us to test a wide variety of spatial relationships.
- They are not as prone to over-fitting as other learning techniques such as maximum entropy models.
- They have no parameters to tune except for the number of rounds they run for and can usually just be run until the test accuracy plateaus [Freund and Schapire, 1999].
- They come with statistical guarantees about the rate of convergence given sufficient data and a set of moderately accurate weak hypotheses [Freund and Schapire, 1999].

Adaboost can be a poor choice as a learning technique if the training data is small, noisy or if there exists a set of data points that no weak hypotheses can split with a even low purity. Outliers that cannot be classified successfully end up with a large weight and the model can become biased towards classifying these excessively hard points. There are variations on Adaboost that compensate for this such as Logitboost [Friedman et al., 2000] which limits the maximum weight of outliers and Brownboost [Freund, 2001] which aggressively down

weights points that are determined to be too hard to classify. Adaboost approaches such as these might be necessary for our work on qualitative spatial classification as some objects in an environment are found outside their normal locations and are very difficult to classify.

2.1.2 Alternating Decision Trees

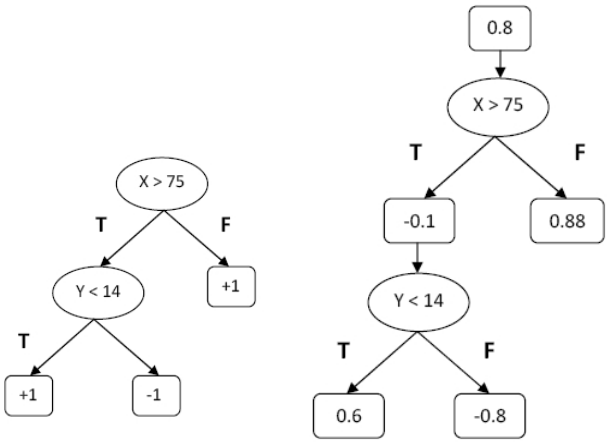
Alternating decision trees (ADTrees) [Freund and Mason, 1999] are a variation of binary classification decision tree that have certain useful properties. They allow for an equally expressive but exponentially more compact representation of any decision tree than the conventional representation shown in Figure 2.1. ADTrees are also considered easier to interpret by a human observer than full decision trees, making it simpler to determine the reasoning behind a decision. The resulting classifier has similar accuracy to other decision tree learning techniques such as C4.5 [Quinlan, 1993] but allows for simpler, more compact tree representation. ADTrees also provide a classification score, a measure of confidence in the resulting classification. Classification scores are not as useful as a probabilistic output because they can not be compared between different types of models but can be used as the basis of a reject option.

Structure

An ADTree is a binary classifier that uses a cumulative scoring value to determine both classification and to provide a classification score. The tree is made up of two types of nodes: decision nodes and predicate nodes. *Decision nodes* contain a binary decision as a predicate condition (e.g., X above Table). *Predicate nodes* contains a classification value (possibly negative) that either raises or lowers the likelihood of the classification. Decision nodes are connected to two predicate nodes, for the positive and negative cases. Predicate nodes can then connect to any number of decision nodes or none. Figure 2.1 shows a conventional decision tree and an equivalent alternating decision tree.

Classification

Like other decision trees, classification in ADTrees begins at the root node and follows the tree structure as dictated by the decision nodes. Unlike regular decision trees where the output is a binary value at the leaf node, ADTrees produce a classification score based on the entire path used to traverse the tree. A running sum over all encountered classification nodes is maintained and the cumulative score is kept along the path from root to leaf.



(a) *Decision Tree* **(b)** *Alternating Decision Tree*

Figure 2.1: *An example of a conventional decision tree and the equivalent alternating decision tree. The circular nodes represent decisions. The square nodes in the decision tree represent classification leaf nodes and in the alternating decision trees they represent predicate nodes.*

The sign of the cumulative score is the classification and the magnitude is a measure of classification confidence.

Training

Major improvements to the accuracy of decision tree classifiers can be achieved by training using boosting and Adaboost has been shown to be an effective basis for improving alternating decision tree classifier construction [Freund and Mason, 1999]. The following is a brief overview of an approach for boosted learning of alternating decision trees and is included to provide an impression of both how ADTrees are trained and how AdaBoost works. This algorithm is for creating a binary classifier but a multiclass classifier can be produced by training N trees for N classes using a 1 vs. all training approach and selecting the classification from the tree with the greatest score.

Let $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be a set of training data where each x_i is a feature vector and $y_i \in \{-1, 1\}$ is a set of labels. C is a set of *conditions*, binary comparisons to the presence of a single feature in an instance x. A *precondition* is a set of conditions needed to reach a specific predicate node. A *rule* is denoted as r and is a combination of a precondition necessary to reach the decision node and the condition at that node. The decision tree is a set of decision node rules R. P is a set of preconditions associated with the possible

locations where a new rule can be added.

Initially, $P_1 = \mathbf{T}$, where $\mathbf{T}$ is a constant *true* predicate representing the root node. Each training point has an associated positive weight $w_{i,t}$ which is the weight for training point i on training round t. Initially, $w_{i,0} = 1$ for all training points i. P_t and R_t correspond to the two sets at boosting iteration t. $W_+(c)$ and $W_-(c)$ represent the sum of all the weighted training values that satisfy condition c and with values of $y = -1, +1$ respectively.

Initialize R_1 with a rule with a precondition and condition equal to $\mathbf{T}$ and with

$$r_1(x) = \frac{1}{2} ln \frac{W_+(\mathbf{T})}{W_-(\mathbf{T})} \tag{2.1}$$

For $t = 1, 2, ..., N$

1. For each precondition in $c_1 \in P_t$ and each condition $c_2 \in C$ find the impurity measure

$$Z_t(c_1, c_2) = 2\left(\sqrt{W_+(c_1 \wedge c_2)W_-(c_1 \wedge c_2)} + \sqrt{W_+(c_1 \wedge \neg c_2)W_-(c_1 \wedge \neg c_2)} + W_+(\neg c_2) + W_-(\neg c_2)\right) \tag{2.2}$$

2. Choose c_1 and c_2 that minimize $Z_t(c_1 \wedge c_2)$. Construct a new rule r_t with precondition c_1 and condition c_2 and prediction values

$$r_t(x) = \begin{cases} \frac{1}{2} ln \frac{W_+(c_1 \wedge c_2)}{W_-(c_1 \wedge c_2)} & \text{if } c_2 \\ \frac{1}{2} ln \frac{W_+(c_1 \wedge \neg c_2)}{W_-(c_1 \wedge \neg c_2)} & \text{if } \neg c_2 \end{cases} \tag{2.3}$$

3. Set $P_{t+1} = P_t + c_1 \wedge c_2 + c_1 \wedge \neg c_2$

4. Update the training same weights $w_{i,t+1} = w_{i,t+1} e^{r_t(x_i) y_i}$

The classification and score for a sample x is found by comparing x against all R_{t+1} rules which is equivalent to traversing the tree.

$$class(x) = sign(score(x)) \tag{2.4}$$

$$score(x) = \sum_{t=1}^{T} r_t(x) \tag{2.5}$$

Determining an appropriate number of training rounds T before training is not generally

possible but since over-training is unlikely, training can continue until test error asymptotes.

In our final experiments, we used a structured SVM learning algorithm rather than a boosted decision tree because our goal was to learn a set of weight associated spatial relationships between true positive detections. This was not a classification problem in the context of decision trees so they were not an appropriate choice.

2.2 Support Vector Machines

Support vector machines (SVMs) are a class of machine learning algorithms introduced by [Cortes and Vapnik, 1995] and used for classification and regression analysis using pattern recognition. Given a set of training data $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where each x_i is a vector in a D dimensional linear vector space and $y_i \in \{-1, 1\}$, SVMs identify a D dimensional hyperplane that can separate this input data into two categories.

SVMs have several useful properties that make them widely used in machine learning.

- SVMs can be applied to a wide variety of problem types with differing input and output spaces.
- The optimization problem that SVMs present is typically convex so finding a unique global optimum can be assured.
- SVMs are efficient to train because the optimization problem required is sparse with respect to the training data.
- Both linear and non-linear solutions can be found using the "kernel trick", a technique to compute the inner product between high-dimensional features corresponding to two points without explicitly computing the high-dimensional features.
- SVMs can handle training data with misclassified data points or other outliers using a soft-s formulation.

Often input data needs to be mapped to a more expressive hypothesis space than its raw representation, so variables or *features* are computed from the training data inputs and a hyperplane is computed which divides the categories in the feature space. A set of features from an example are called a *feature vector* and the vectors that lie close to the hyperplane are called *support vectors*.

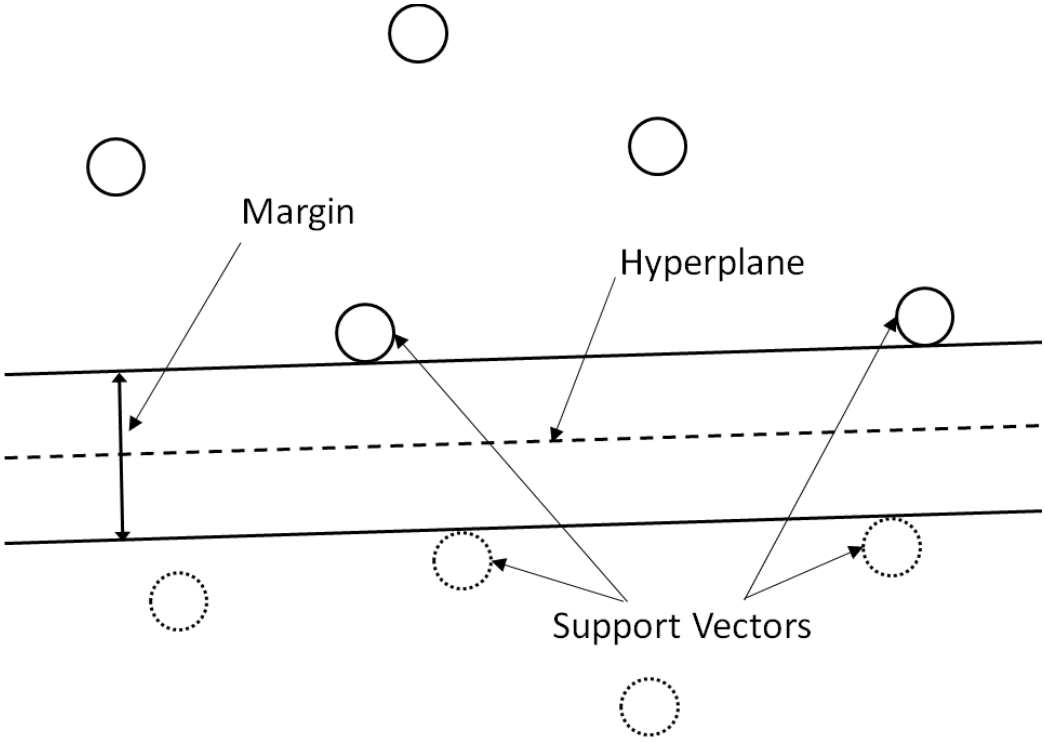


Figure 2.2: *A visualization of a 2D SVM classifier. The hyperplane is computed between the solid and dotted line data points.*

2.2.1 Linear Support Vector Machines

A hyperplane can be defined by the equation $\langle w, x\rangle + b = 0$ where $(w, b) \in \mathbb{R}^M \times \mathbb{R}$. In a SVM, the hyperplane is optimized to maximize the distance from the plane to the support vectors from both categories. This distance is called the *margin* and, generally, the larger the margin between the support vectors and hyperplane, the lower the error in classification. The following overview of SVM classification is based on [Smola and Schölkopf, 2004].

A margin is defined as:

$$\gamma_i = y_i(\langle w_i, x_i\rangle) + b) \tag{2.6}$$

and $\gamma > 0$ indicates a positive classification. See Figure 2.2 for a visualization of the SVM hyperplane and margin.

The *geometric margin* is a normalized version of the margin used to measure the Euclidean

distance of the margin,

$$\gamma_g i = \frac{y_i(\langle w_i, x_i \rangle) + b)}{\|w\|} \tag{2.7}$$

We can pose the computation of the SVM as an optimization problem where the objective is to maximize the geometric margin.

$$\begin{array}{ll} \max_{w,b} & \gamma_g \\ s.t. & \frac{y_i(w\dot{x}_i+b)}{\|w\|} \leq \gamma_g, i = 1, \ldots, l \end{array} \tag{2.8}$$

The geometric margin is equal to $\frac{1}{\|w\|_2}$ (proof omitted), so the optimization problem can be reformulated as

$$\begin{array}{ll} max_{w,b} & \frac{1}{2}\langle w, w \rangle \\ s.t. & y_i(w\dot{x}_i + b) \leq 1, i = 1, \ldots, l \end{array} \tag{2.9}$$

This optimization problem assumes that the results are linearly separable and there is no misclassified training data but that limitation will be dealt with later. In order to optimize this equation and employ the kernel trick later, it is necessary to give the problem a dual representation. In many optimization problems, it is often useful to convert the primal (original) representation of a problem into a *Lagrangian dual problem* because the solution to the dual problem provides a lower bound to the solution of the primal problem. To find the dual representation, we first need the Lagrangian primal representation.

For an optimization problem of the form

$$\begin{array}{ll} \min_x & f(X) \\ s.t. & g_i(X) \leq 0, i = 1, \ldots, j \\ & h_i(X) = 0, i = 1, \ldots, k \end{array} \tag{2.10}$$

the Lagrangian primal form is,

$$L(X, \alpha, \beta) = f(X) + \alpha g(X) + \beta h(X) \tag{2.11}$$

where α_i and β_i are called *Lagrangian multipliers.*

Using this definition, the Lagrangian primal form of Equation 2.9 is

$$L(w, b, \alpha) = \frac{1}{2}\langle w, w \rangle - \sum_{i=1}^{l} \alpha_i [yi(\langle w_i, x_i \rangle + b) - 1] \tag{2.12}$$

Using the Kuhn-Tucker conditions [Kuhn and Tucker, 1951], the dual Lagrangian problem can be determined for the primal. The dual problem takes the form of the quadratic

optimization

$$\begin{array}{ll} max_\alpha & f(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle x_i, , x_j \rangle \\ s.t. & \sum_{i=1}^{l} y_i \alpha_i = 0 \\ & \alpha_i \geq 0, i = 1, \ldots, l \end{array} \tag{2.13}$$

Once $\max_\alpha$ has been computed, the max-margin hyperplane determined by (w^*, b^*) can be computed by

$$\begin{array}{l} w^* = \sum_{i=1}^{l} y_i \alpha_i x_i \\ b^* = \frac{1}{2}[\max_{y_i=-1}(\langle w^*, x_i \rangle) - \min_{y_i=1}(\langle w^*, x_i \rangle) \end{array} \tag{2.14}$$

2.2.2 Non-linear Support Vector Machines

Often training examples are not linearly separable in the initial problem space, so SVMs map the input data points into a higher dimensional space within which they are linearly separable. However, in a higher dimensional feature space, computing the SVM is more computationally expensive. SVMs use a technique *kernel trick* [Aizerman et al., 1964] to avoid this problem which allows the algorithm to fit a maximum margin hyperplane without needing to compute the higher dimensional features explicitly. For a full examination of non-linear support vector machines and the application of the kernel trick see [Burges, 1998].

2.2.3 Structured Support Vector Machines

Our model in Chapter 7 uses a technique called Structured Support Vector Machines [Tsochantaridis et al., 2004] to learn a set of weights associated with inter-object spatial relationships. Structured output prediction is an area of machine learning concerned with the prediction of complex, structured outputs. Structured SVMs learn a discriminative function from X to discrete outputs Y, where Y is a complex output that could be sequences, strings, labeled trees, graphs, etc. In regular multi-class SVMs, the output space is a set of labels $Y = \{1, \ldots, K\}$. Naively, a regular multi-class SVM can be used to produce complex outputs by treating each possible state as a label. However, this approach would likely result in a very large and complex output space and computing a SVM in that space is infeasible to solve. Structured SVMs solve this problem by taking advantage of dependencies within Y and a max margin algorithm to learn the weights w.

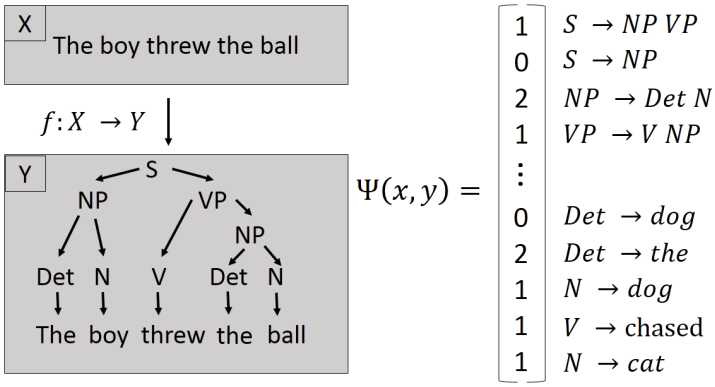


Figure 2.3: *An example of an input x, output y and feature map $\Psi(x, y)$ for a structured SVM designed to determine a Probabilistic Context Free Grammar. Figure reproduced from [Tsochantaridis et al., 2004]*

Problem Formulation

Structured SVMs learn a *discriminant function* $F : X \times Y \to \mathbb{R}$ over input/output pairs. Prediction is performed by maximizing F over the response (output) variable for an input x. The hypothesis f is then

$$f(x; w) = \arg\max_{y \in Y} F(x, y; w) \tag{2.15}$$

where w is a parameter weight vector. F is assumed to be linear in $\Psi(x, y)$ which is a combined feature representation of input and outputs,

$$F(x, y; w) = \langle w, \Psi(x, y) \rangle \tag{2.16}$$

where $\Psi(x, y)$ will have different forms depending on the problem space. The structure of $\Psi(x, y)$ has significant impact on the complexity of the problem. In Figure 2.3 we provide an example from [Tsochantaridis et al., 2004] of the X, Y and $\Psi(x, y)$ for the problem of learning a Probabilistic Context Free Grammar from natural language processing using a structured SVM. In this example, the problem is determining a parse tree y for each sentence x using a set of grammar rules g. $\Psi(x, y)$ for this problem is a histogram count of how often each grammar rule occurs in tree y for sentence x.

Using structured SVMs requires $\Delta(y, \hat{y})$, a loss function that quantifies the difference between prediction $\hat{y}$ and the true output y. Training any SVM requires a loss function but

using a simple zero-one loss function for comparing structured outputs typically does not work well. A loss function is required that can capture whether an output is close to correct or completely wrong.

Given a problem space and loss function where $\Delta(y, y') > 0$ *s.t.* $y \neq y'$ and $\Delta(y, y) = 0$, we can formulate the problem with zero training error as a set of non-linear constraints

$$\forall i : \max_{y \in Y \setminus y_i} \{\langle w, \Psi(x_i, y)\rangle < \langle w, \Psi(x_i, y_i)\rangle\} \tag{2.17}$$

This in turn can be rewritten at a set of linear constraints

$$\forall i, \forall y \in Y y_i : \langle (w, \delta\Psi_i(y)\rangle > 0 \tag{2.18}$$

where $\delta\Psi_i(y) = \delta(x_i, y_i) - \delta(x_i, y)$.

Given that there can be many possible solutions w^* for the constraints in Equation 2.18, Tsochantaridis *et al.* propose selecting w where $\|w\| \leq 1$ and the score of the correct label y_i is most different from the next closest. This step incorporates the max-margin principle from general SVMs into the problem of producing a structured output. The resulting formulation is the following:

$$\begin{gathered} SVM : \min_w \tfrac{1}{2}\|w\|^2 \\ \forall i, \forall y \in Y \setminus y_i : \langle (w, \delta\Psi_i(y)\rangle \leq 1 \end{gathered} \tag{2.19}$$

This formulation does not allow for training error, so slack variables are introduced to optimize for a soft-margin criteria. Also, Equation 2.19 is for the zero-one classification case and does not incorporate the loss function $\Delta(y, y')$. There are two approaches to incorporate slack variables and $\Delta(y, y')$. This first is slack rescaling which follows the model proposed by [Crammer and Singer, 2002] where a slack variable is introduced for every non-linear constraint and then rescaled according to the loss from each constraint,

$$\begin{gathered} SVM : \min_w \tfrac{1}{2}\|w\|^2 + \tfrac{C}{n}\sum_{i=1}^{n,\varepsilon} \varepsilon_i \text{ s.t. } \forall i \varepsilon_i \geq 0 \\ \forall i, \forall y \in Y \setminus y_i : \langle w, \delta\Psi_i(y)\rangle \leq 1 - \tfrac{\varepsilon_i}{\Delta(y_i, y)} \end{gathered} \tag{2.20}$$

The second approach is margin rescaling by [Taskar et al., 2003] which adjusts the margin constraints. The margin rescaling formulation of the structured SVM problem is

$$\begin{gathered} SVM : \min_w \tfrac{1}{2}\|w\|^2 + \tfrac{C}{n}\sum_{i=1}^{n,\varepsilon} \varepsilon_i \text{ s.t. } \forall i \varepsilon_i \geq 0 \\ \forall i, \forall y \in Y \setminus y_i : \langle w, \delta\Psi_i(y)\rangle \leq \Delta(y_i, y) - \varepsilon_i \end{gathered} \tag{2.21}$$

Learning

The potentially very large number of constraints associated with solving a structured SVM presents a challenge. The algorithm presented in [Tsochantaridis et al., 2004] to solve this problem is called called the *Cutting Plane Method* which iteratively finds a small set of constraints S_i for each training example. Constraints are added by repeatedly identifying the most violated constraints for each training example for intermediate solutions and adding them to S. The intermediate solution weights w are determined by α_{iy}, a Lagrange multiplier enforcing the margin constrains from S using standard quadratic programming techniques.

The following is the pseudocode for the cutting plane method used in both [Tsochantaridis et al., 2004] and the StructSVM toolkit [Vedaldi, 2011], for computing a structured SVM:

Algorithm 1 Algorithm for computing a Structured SVM for both slack and margin rescaling

Precondition: $(((x_1, y_1), \ldots, (x_n, y_n)), C)$
1: $S_i \leftarrow \emptyset$ for all $i = 1, \ldots, n$
2: **repeat**
3: **for** $i = 1, \ldots, n$ **do**
4: $w \leftarrow \sum_j \sum_{y' \in S_j} \alpha_{jy'} \delta\Psi_j(y')$
5: **if** using slack rescaling **then**
6: $SVM^{\Delta s}: \ H(y) = (1 - (\delta\Psi_i(y), w))\Delta(y_i, y)$
7: **else**
8: $SVM^{\Delta m}: \ H(y) = (\Delta(y_i, y) - (\delta\Psi_i(y), w))$
9: **end if**
10: Compute $\hat{y} = \arg\max_{y \in Y} H(y)$
11: Compute $\varepsilon_i = \max(0, \max_{y \in S_i} H(y)$
12: **if** $H(\hat{y}) > \varepsilon_i$ **then**
13: $S_i \leftarrow S_i \cup \{y_i\}$
14: $\alpha_S \leftarrow$ optimize dual over $S, S = \cup_i S_i$
15: **end if**
16: **end for**
17: **until** no S_i has changed over entire iteration

The algorithm for both slack and margin rescaling are the same except at step 5 where there functions are selected. In step 10, the algorithm determines the most violated constraint for an output $\hat{y}$. Efficiently determining the most violated constraint is potentially very computationally intensive and, since it is based on the cost function $H(y)$, different approaches are necessary for the slack and margin rescaling solutions.

Implementation

Tsochantaridis *et al.* provide code for the learning structured SVMs in a toolkit called StructSVM [Vedaldi, 2011]. This toolkit allows for the implementation of a structured SVM given a problem formulation defined by three functions. First is a loss function $\Delta(y, y')$ which should encode loss in a more sophisticated manner than simply zero-one loss. The second is a feature mapping function $\Psi(x, y)$ which maps the combined input and output space onto a joint feature space. Lastly is the maximization step performed in step 10 of the pseudocode algorithm which finds the most violated constraints.

Chapter 3

Context and Object Detection

The overall goal of this chapter is to ground our work in the existing literature of object detection using context. This chapter describes a number of approaches to performing object detection with the aid of context and contrasts them with our work. We have divided the chapter into two sections. First we describe a number of approaches that explicitly use 3D contextual information either about the structure of the scene or the layout of the objects to perform object detection or similar tasks. The second section describes a number of 2D object detectors that perform joint detection of objects in the scene using context.

3.1 Image-based Object Detection

An *object* is a physical "thing" in an environment with a *semantic label* (i.e., a word which places it in a category). Objects can be separable from the environment or permanently attached elements. Semantic labels provide a way of clustering different objects into a *category* or *class* (things with the same label) and a way of decomposing a scene into objects (things with different class labels). Each object has an associated region, the pixels in an image that correspond to that object.

A class is an abstract grouping of individuals, other classes or both, also referred to as category, sort, or type. The generic term "Object" is often used as the root of a *taxonomy*, a hierarchy where each level describes a logical grouping of entities based on shared physical properties, common uses, semantic usage, or other similarity metrics. Belonging to the object class entails that objects exhibit certain properties, such as occupying a region of space. Taxonomic hierarchies are based on an "'is a"' relationship (e.g. X is a either equal to or a child node of Y if X is a Y). These groupings serve to both define the properties of an object and differentiate it from other objects.

Object recognition is the general problem in computer vision of detecting objects or object classes in an image. It is a term used to describe a number of different types of problems.

Object recognition problems focus on the task of identifying one or more target object classes. *Object Classification* is the binary classification problem of determining whether an image or part of an image contains a target object. If there are multiple target object types, then the problem is called *Object Categorization*. *Object Localization* is the process of determining regions within an image for each occurrence of the target object. *Object Detection* is the problem of both classifying/categorizing and localizing a target object type(s) in an image.

3.1.1 Object Detectors

Most detectors produce two outputs for a positive object detection: a region and a confidence score. *Object regions* are the results of object localization, generally represented as rectangular *detection boxes* or pixel-based *image masks*. The *confidence score* associated with a detection is a measure of how closely the object's appearance in the image matches the model used for detection. Depending on the model, this score may be derived from a margin of classification for support vector machines (SVMs) or a probability for probabilistic models. Other models simply have the property that there is a monotonic relationship between score and the likelihood that a detection is correct. Object detectors designed for a broad class of objects with significant variance in appearance (e.g., humans, chairs, cars, etc.) are called *generic object detectors*.

Most object detectors use image features as the basis for the model. *Image features* represent a portion of an image (corresponding to an object) in a manner which is invariant to changes in illumination, orientation and viewpoint. This invariance is important if an object detector model is to generalize from its training data to test data. Examples of successful image features include the Scale Invariant Feature Transform (SIFT)[Lowe, 2004] and the histogram of gradients [Dalal and Triggs, 2005]. These features are based on image gradients but image features have also been based on the silhouette or contours of objects [Shotton et al., 2008], [Helmer and Lowe, 2010]. For a broad survey and comparison of many image features and a comparison of different object detection approaches see [Zhang et al., 2007].

In this work, we use object detectors designed to recognize man-made objects in indoor scenes, in contrast to detectors designed to identify natural objects (e.g., trees, animals, etc) or background scene elements (e.g., roads, sky, grass, etc). Most object detectors represent the objects as a collection of image features, with either a sparse or dense representation. *Part-and-shape object detectors* break down the object detection problem into multiple detection problems for constituent object parts and they model the spatial distribution of

those parts. These models are technically more challenging but have shown great success at object detection [Fergus et al., 2005], [Felzenszwalb and Huttenlocher, 2005], [Felzenszwalb et al., 2008].

Sparse representation object detectors are the most widely used approach to object detection [Zhang et al., 2007] and work by identifying a small set of consistent visual patterns in the target object and representing objects as a collection of image features. Sparse object detectors are computationally efficient because they function in a sparse feature space rather than trying to model the dense pixel information. Many early approaches to sparse object detection used orderless *bag-of-features* models [Grauman and Darrell, 2007; Zhang et al., 2007], which have origins in text recognition. In this model, image features are first extracted from object regions. From these, a "codebook" of highly discriminatory features is learned and histograms or other counts of these features are used for object detection. This technique can be improved using *spatial pyramids* where feature are computed over image cells defined by a recursive multi-level image decomposition [Lazebnik et al., 2006]. Orderless bag of feature models are limited in their ability to perform object detection by the lack of structural information about the layout of features and have generally been more useful at image classification.

Dense representation object detectors densely model the appearance of objects. Detection can be performed at the pixel-level (e.g., template matching) or through densely computed image features [Dalal and Triggs, 2005]. For most generic object detection, using a dense rigid template to model the appearance of an object is in effective because appearances vary too much within the class. However, combining rigidly computed feature based matching with part-and-shape based detection has lead to very effective generic object detectors [Felzenszwalb et al., 2008], [Bourdev et al., 2010].

3.1.2 Discriminatively Trained Deformable Part Model

In our work we use an image-based object detector to produce our 3D object detections. The discriminatively trained deformable part model (DPM) [Felzenszwalb et al., 2008] is a highly effective object classifier which has shown significant success in the Pascal Visual Object Classes Challenge [Everingham et al., 2010]. It is a part-and-shape based model that uses a coarse global template that matches the entire object and multiple higher resolution part templates.

The underlying feature used for the templates is the histograms of oriented gradient features (HOG) [Dalal and Triggs, 2005]. HOG decomposes an image region into a grid of cells,

computes a histogram of image gradients within those cells, produces normalized histograms of those gradients and returns a feature for each cell. The HOG features are computed over an image pyramid to capture both coarse and fine level object features. The HOG features are matched against the object model using filters which specify weights for subwindows of the HOG pyramid. The score of a filter is the dot product of the weight vectors and the features in the subwindow of the HOG pyramid. An object model then consists of a root filter F_0 which captures the entire object and n part filters $(P_1, \ldots, P_n)$ where $P_i = (F_i, v_i, s_i, a_i, b_i)$. Here F_i is a part filter, v_i and s_i are the center and size of a box of possible part positions relative to the root, and a_i and b_i specify the coefficients of a quadratic function which measures the score of the placement of P_i.

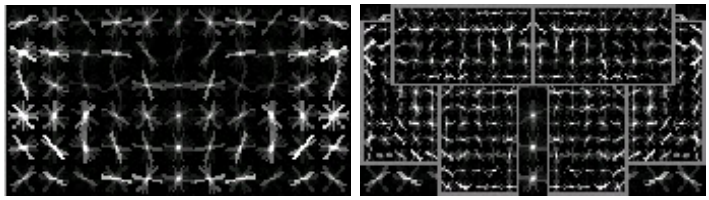
The weights used for scoring are learned from a training set of images annotated with the type and bounding box for each object type being detected. The task is a binary classification given an input set of labeled examples $D = ((x_1, y_1), \ldots, (x_n, y_n))$. x_i is a HOG pyramid $H(x_i)$ and a set of valid locations for a root filter $Z(x_i)$ and $y_i \in -1, 1$ is a set of labels. Positive and negative training sets are constructed from this data with the negative set containing only images where there are no instanced of the classification object. The underlying model is called latent SVM, which is a reformulation of multi-instance SVM in terms of [Andrews et al., 2002] latent variables. The approach alternates between fixing latent values for positive examples and optimizing the SVM object function.

The approach commonly used with the DPM is to learn multiple object classifiers, each of which is designed to capture the object type either from a different viewpoint or capture significant variations in structure within the class. This is done by sorting and then splitting the training data set according to the ratio of the height and width of the object boxes. Figure 3.1 shows a visualization of the two models learned for a bowl classifier.

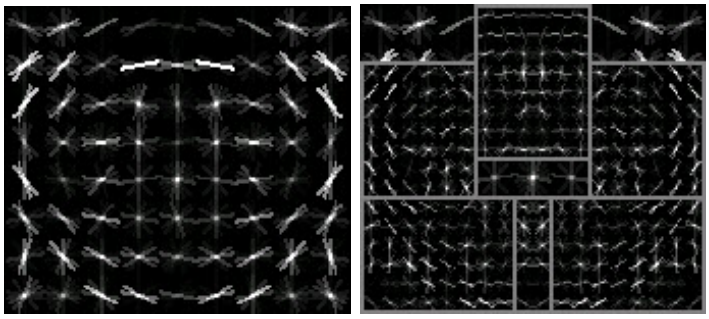
Detection is performed using a scanning windows approach. The score for a detection is derived from the score of the overall template matched against the window and the sum of all the part templates minus a deformation cost. The scores are computed by the dot product of the histogram of gradients and a learned set of weights.

3.2 Context in Object Detection

While it is agreed upon in both the psychological and computer vision communities that context is valuable for object recognition, defining context is a challenge. Often it is left undefined and implicitly refers to whatever effects the researcher is examining that influences the objects in a scene or image. Context was defined by [Strat, 1993] as “any and all



(a) *First component of bowl* **(b)** *First component of parts model (corresponding to a side-model of bowl ways view)*



(c) *Second component of bowl* **(d)** *Second component of parts model (corresponding to a topmodel of bowl down view)*

Figure 3.1: *Visualizations of the Felzenszwalb et al. object classifier. The images on the left show an expected intensity of gradient in a grid pattern for the entire object. On the right, they show the gradients in the parts model.*

information that may influence the way a scene and the objects within it are perceived". This seems an overly broad definition as it would include object appearance in context so we define *context* as "all information that affects the appearance or layout of a scene or the objects it contains".

[Divvala et al., 2009] published a broad study of the effects of context on object detection in images and created a taxonomy of contextual effects and types of contextual information which we use as the basis for discussing context. We have expanded this to consider context in 3D environments as well as images. The following is a list of the effects that context has on objects which influence this thesis.

- *Object Presence:* Context affects whether an object is present within a given area. In order to determine object presence the area of consideration needs to be defined. For images the area is defined by the boundaries of the image. In 3D the area needs to be defined and could, for example, be a room, a building, or a semantically defined area like a kitchen or office. Information about the type of scene provides a strong prior

on the types of objects likely to be found. The presence or absence of other objects in a scene or image also has a contextual effect on object presence (e.g., toilet presence is correlated with bathtubs but not with microwaves).

- *Object Position:* Context affects where objects are located. Object positions are influenced by many factors such as the type of scene, the structure of the immovable scene elements (e.g. walls, floors, work surfaces, etc), the position of other objects and the areas where activities are typically performed. In this thesis, we are particularly concerned with the way object positions are influenced by other object positions. As a source of contextual information, object positions provide insight into both the immovable elements of the scene and the areas where activities are performed.

- *Object Appearance:* Context affects the way an object's appearance changes in different scenes. Appearance is influenced by factors such as illumination, weather and the viewpoints from which information about the scene are acquired. Context can also influence the physical structure (and therefore appearance) of a class of objects in different scene types. For example, a chair in an office environment is physically very different than one in a kitchen or on an airplane.

Our work is primarily concerned with the effects of context on object position but object presence and appearance are both factors that we must consider. In object recognition challenges like the PASCAL Challenge [Everingham et al., 2010], object presence can have a significant effect because the objects in the challenge come from many different types of scenes [Desai et al., 2009] and for each scene there are few objects likely to be found. However, in the model we present in Chapter 7 we already know the scene and are only searching for objects commonly found in that scene type. Not being able to use object presence as a factor in improving object detection accuracy makes our work more challenging but our goal is to demonstrate the effects of object position, not object presence.

We primarily considered the effect of object appearance in our selection of training data. For example, when training our chair classifier, we used the query "office chair" rather than simply "chair" in order to get images more like the objects in our training and test data. Other contextual appearance effects, like illumination, also affect the objects in our scenes. There are significant differences in lighting, and therefore appearance, between objects in real world scenes and objects in commercial product shots. To accommodate this, when selecting images for training our object detectors, we ensured that a variety of different sources with different illumination conditions were used. This leads to an object detector able to accommodate different lighting conditions.

3.2.1 Contextual Sources

[Divvala et al., 2009] also created a taxonomy of contextual information sources. We found their taxonomy very broad and complete but there is no clear motivation for the division of their categories. Some categories were based on the cause of the context information (e.g., weather, culture, 3D scene structure, etc) while others were based on how that contextual information is manifested in the image (e.g., local pixel effects, 2D global statistics, etc). The following list is a modified version of their taxonomy we use in this thesis, where all the categories are based on the cause of the context.

- *Scene Context* is contextual information based on analysis of the scene as a whole. The type of scene is a strong contextual information source. Context can be provided by low-level global image statistics of the scene, often referred to in computer vision literature as *gist*, [Oliva and Torralba, 2001], [Russell et al., 2007] which learns contextual effects without explicitly classifying the scene. Other approaches use context derived from explicitly determined scene type or the use of keywords associated with the scene or image [Li and Li, 2010], [Oliva and Torralba, 2001], [Carolina Galleguillos and Lanckriet, 2010]. Scene context also includes context provided by activities present in the scene.

- *3D Geometric Context* is contextual information based on the 3D scene structure. This includes detection of supporting surfaces, contact points, visual occlusions and alignment with walls or other scene structures [Hoiem et al., 2006].

- *Object Context* is contextual information based on object-object interactions. This include object presence and object position effects in 2D [Galleguillos and Belongie, 2010], [Carolina Galleguillos and Lanckriet, 2010], [Galleguillos et al., 2008],[Rimey and Brown, 1994], [Neumann and Möller, 2007] or 3D [Hoiem et al., 2007], [Anguelov et al., 2005]. Object context in 3D is the main source of context we examine in this thesis and will be discussed in greater detail in Chapter 5.

- *Photographic Context* is contextual information based on an image's intrinsic properties and how it is acquired. This includes the camera's height, focal length, orientation, and other internal properties. It also includes cultural effects which influence how the operator targets the camera such as the common position and pose of people in images, the effects of framing [Simon and Seitz, 2008], and the selection of culturally appropriate subject matter. Photographic context also includes effects resulting from how a robot acquires images [Meger et al., 2008]. Some of these photographic con-

textual effects also influence 3D scenes that is acquired from images or from scanning devices that are targeted like a camera.

- *Environmental Context* is contextual information based on the outdoor environment of the scene. This includes the effects of lighting (both natural and artificial)[Lalonde et al., 2008], weather [Narasimhan and Nayar, 2002], and terrain types [Hays and Efros, 2008]. It is similar to scene context but the effects come from sources a long way away from the scene (sun, clouds, distant terrain, etc).

- *Temporal Context* is contextual information based on temporal information about the scene. Temporal information about the events before and after the image can be determined from a video or a set of sequential images [Liu et al., 2008].

In order to focus the taxonomy on only contextual sources and incorporate 3D scenes as well as images, we made the following changes from the original taxonomy in [Divvala et al., 2009]. We removed "Local Pixel Context" which combined effects from many sources that manifested locally in the image. We combined "2D Scene Gist Context" and "Semantic Context" into the more category of "Scene Context" since both are broad analysis of the scene as a whole. We combined "Illumination" and "Weather" and "Geographic" context into the more broad category of "Environmental" context because of potential overlaps in contextual sources (e.g., is smog weather or geography and is the effect of city street lights illumination or geography). We also combined "Cultural" context with "Photogrammetric" context into the "Photographic" category because it is unclear whether effects like framing or intentionally blurred backgrounds in photos had photogrammetric or cultural origins.

Generally, all of these contextual sources have some influence on all the contextual effects: object presence, position and appearance. They are complementary sources of information, though sometimes one dominates another. For example, from the objects in a scene, it is often possible to to determine scene type, as we will discuss in Chapter 4. Our work focuses on object context but object context can provide 3D geometric context because from the relative positions of objects in a scene it is possible to infer scene structure.

3.3 Object Detection Using 3D Context

The most significant element of our work that differentiates it from previous work on context in object detection is the use of 3D data and spatial relationships. We are not the first to explore how 3D data can effectively be used to improve object classification. This section

covers several different examples of research that have employed context in 3D environments for object detection and how they are applicable or affect the work presented in this thesis.

3.3.1 Context and the Spatial Semantic Hierarchy

The work of Kuipers *et al.* has long focused on the problem of a robot embedded in a rich 3D environment, attempting to bootstrap knowledge about places, objects and actions. [Kuipers et al., 2000] introduced the concepts of the Spatial Semantic Hierarchy, a model with multiple interacting representations of spatial knowledge for robots. The model combined qualitative and quantitative representations of space. Of particular interest to us are the upper-level, abstracted layers that decomposed space along high-level concepts of paths, places, and regions. In [Kuipers et al., 2000], objects had a limited role in the spatial semantic hierarchy model but they expanded on this in their work on autonomous object discovery [Modayil and Kuipers, 2004]. Their approach separated objects from their surroundings, using 3D geometric contextual information both to discover new objects and produce object detection models from 3D range data. They combined concepts from their spatial semantic hierarchy and object discovery into the Object Semantic Hierarchy [Xu and Kuipers, 2010]. This multilayer approach discovers objects by modeling the contextual surrounding, identifying new 2D views of objects and using them to produce 3D models of objects.

The work of Kuipers *et al.* demonstrates well how embedded robots can reason about their surrounding, learn object models in both 2D and 3D and use those models to automatically discover structure and objects. Their work on object discovery was influential in how we think about leveraging context for both object discover and to identify new information about known objects. They use context to improve object detection by performing better object-background segmentation and using this information to create better models. Our approach differs significantly from theirs in our greater emphasis on spatial relationships, our use of web-acquired training data for object recognition and less emphasis on the specifics of robotic implementation.

3.3.2 Monocular 3D Context

The authors of the study on context we used as the basis of Section 3.2 [Divvala et al., 2009] have demonstrated an approach to object detection that creates a coarse 3D model of the scene computed from a single image from an uncalibrated camera and uses the 3D contextual information to improve object detection [Hoiem et al., 2006]. Their approach

computes object spatial properties (height and scene depth) using camera height and the horizon line. Using an image-based object detector, they detected object candidates in the image. They then used the estimated heights to recompute the candidate's likelihood using a probabilistic framework. We adapt this approach of computing detections using standard object recognition techniques, augmenting them with 3D information and adjusting them to remove false positives. Their work is restricted by using only a single image which meant the 3D information limited and they were only able to detect objects on the ground plane. They also did not use any object-object relationships, likely because they were only applying their detector to a small number of object classes (cars and pedestrians).

Subsequent work by them, [Hedau et al., 2009], has significantly improved on their ability to determine 3D structure of the scene in terms of object positions. They demonstrated an impressive level of accuracy in computing the structure of fixed elements of a scene and approximating the 3D shape of the scene objects. This work has progressed to allow them to compute the support relationships between objects [Silberman et al., 2012]. Others have had similar success at determining the 3D layout of scenes from single images. [Lee et al., 2010] and [Tsai et al., 2011] have demonstrated that a robot can, with a single image, extract information about the 3D structure of the fixed structural elements of the scene. [Fouhey et al., 2012] demonstrates how human pose estimation can be used to further improve the accuracy of single image 3D geometric reconstruction by relating poses to underlying 3D scene structures (e.g., a seated human is likely on a surface with a predictable height).

Single image 3D reconstruction has bearing on our research as our techniques for using 3D spatial relationships could be combined with this approach to perform improved object detection in 3D using a single image. However, it is unclear, given the prevalence and diminishing cost of 3D sensors, how necessary determining 3D structure from single images will be for robots.

3.4 Multi-object Simultaneous Object Detection

This section covers techniques that employ context in the simultaneous detection of multiple objects simultaneously. These techniques generally work by combining both object detection results with contextual information between detection to improve the overall object detection results. [Hedau et al., 2009]

3.4.1 Graphical Models for Contextual Object Detection

In our work, we needed to decide on an underlying model for improving object detection using spatial relationships. *Probabilistic graphical models* are a widely used AI technique and are an efficient and straightforward way of specifying complex probabilistic relationships between variables and can be used to find the probability of an event given varying types of evidence [Torralba et al., 2010]. This section examines some examples of how graphical models are used in 3D contextual object detection.

Some of the earliest work that employed graphical models to apply context to object classification was [Rimey and Brown, 1994] which used selective perception for scene classification. They used a Bayesian network model of a tabletop to determine where a camera should focus to classify the scene efficiently. Their scene was very simple, a table set for a tea party with a small range of object types, and the 2D spatial relationships between the objects were provided, not learned.

More commonly used graphical models for incorporating context into object detection include Markov random fields (MRFs) or conditional random fields (CRFs), both types of discriminative undirected probabilistic graphical models. For further details on MRFs, CRFs and their use in object detection see [Li, 2009]. Both MRFs and CRFs are often used in vision tasks where the nodes represent values with a spatial arrangement such as pixels, image regions, or mesh points and allow for the simultaneous detection of different object classes. In low level approaches, the nodes are typically only locally connected to their immediate neighbors, so it is difficult for contextual information to effectively propagate. This limitation can be overcome by adding edges between more distant nodes or including additional nodes to represent higher level concepts like entire objects [Shotton et al., 2009; Torralba et al., 2004]. [Winn and Shotton, 2006] use a shape-and-part based detection approach, with one layer of nodes representing object parts and another connected layer for detecting entire objects. Their approach also incorporates some 3D information with incorporating edges that model the occlusion and overlap of objects in the scene.

One of the 3D object detection problems where graphical models are most often employed is *mesh segmentation*, the problem of detecting objects or scene structures in a triangle mesh or 3D point cloud. These meshes or point clouds can be generated by laser range finders or computed from stereo techniques and can the 3D points can include intensity or color information. For mesh segmentation, MRF nodes often correspond to mesh points with node potentials computed from a local 3D shape descriptor at that point (e.g., spin images [Johnson and Hebert, 1999] and shape context [Belongie et al., 2002]). The edge weights represent interactions between local mesh points and might include distance or difference

in surface normal. The use of 3D contextual information can include height from ground and global scene shape statistics [Kalogerakis et al., 2010].

3.4.2 Automatic Place Modeling Using Objects

[Ranganathan and Dellaert, 2007] which demonstrated an approach for performing automatic place modeling with a robot, has been influential on this thesis. Their problem is, given an image and an associated depth map of a scene, how can a robot infer a label for the scene and label the types of observed objects and their 3D locations. They used a generative graphical model for this problem learned from supervised training data. This resembles the problem we solve in Chapter 7, the detection of multiple objects based on their appearance and spatial layout, but their emphasis is on the easier problem of place recognition rather than object detection. In our work we are trying to identify the types of objects based on our estimates of their 3D locations, assuming the scene label is already known. Figure 3.2 shows their graphical model which connects pixel level information on appearance and depth to place categorization. [Ranganathan and Dellaert, 2007] used a Markov random field over color/depth pixels with edges connecting to their closest neighbors in order to spatially clusters features and learn a correspondence from pixel features to object types.

One of the most novel elements of this work is the use of 3D object position for both object detection and place categorization. To describe the spatial relationships of objects in the scene, they used a variant of the *constellation model*, a commonly used model in part-and-shape object detectors that encodes the distribution of parts (or in this case objects) in terms of their positions relative to a base location. Using the constellation model required the assumption that all object positions were conditionally independent of each other. This differs significantly from our work where we are trying to demonstrate that object-object relationships are very effective for improving object detection. Their reason for assuming conditional independence of object position is that a graphical model which considered object-object relationships in 3D would would be too complex to produce a viable model. Their concern is using object-object spatial relationships would require a fully connected graph of objects (i.e., the type and position of all objects would influence all others) which does not scale. Though they stated that this was preliminary work and that they hoped to refine it further, the lack of subsequent work on this model suggests that they may have encountered problems extending it. Instead, in their later work they moved away from classifying objects and instead towards global image features [Ranganathan and Lim, 2011].

Though clearly ambitious and impressive work, the limitations they discussed are a contributing factor to why we decided against using a graphical model for this thesis and instead

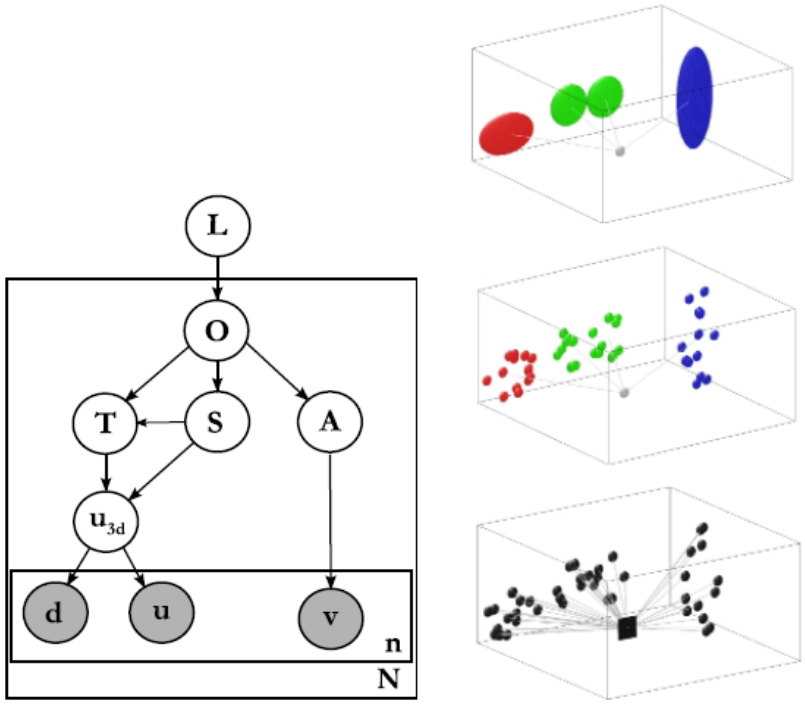


Figure 3.2: *The graphical model for place recognition using object detections in 3D from [Ranganathan and Dellaert, 2007]. The place label L generates a set of N object detections O. Each object detection has a 3D position T, a shape S and an appearance A. The position and shape produce a set of 3D points μ_{3D}. These points are computed from an image containing n features with each feature having a depth d, a pixel location u and an appearance v. Figure reproduced from [Ranganathan and Dellaert, 2007]*

opted for the structured SVM approach for improve object detection that we will discuss in the next section. Their representation of object spatial relationships is simple and they felt that a more complex model would have been infeasible. In our work, we wanted to be able to test a large number of spatial relationships to examine their comparative effectiveness, without needing to be overly concerned about model complexity. For this reason, we decided to avoid the use of graphical models and instead focused on the work of [Desai et al., 2009] and their use of structured SVMs to improve object detection accuracy.

3.4.3 Bayesian Compositional Hierarchy

In many contextual detection problems , the distribution of objects has a hierarchical structure. Consider the example of tables in a restaurant. At the high level the layout of the tables can have a predictable structure, at a lower level the layout of chairs and place

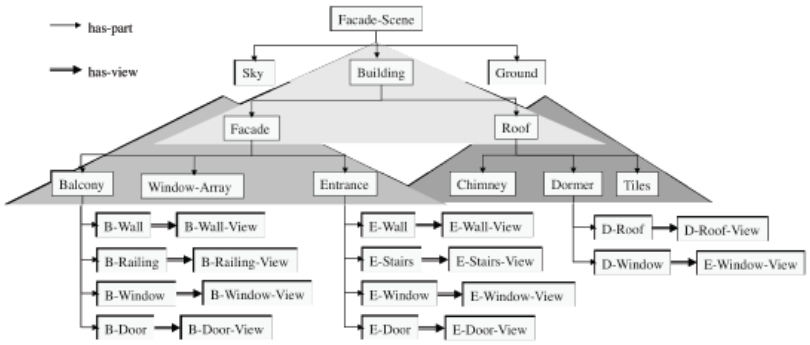


Figure 3.3: *The compositional hierarchy of a facade. Triangle indicate aggregate structure. Figure reproduced from [Terzić and Neumann, 2010]*

settings relative to each table has structure, and at the lowest level the arrangement of utensils in the place settings has structure. While it would be technically possible to use a model which compared the layout of every utensil to every other in the scene, a more logical approach is to break the scene up into components in a hierarchy and learn relationships at each level of the hierarchy. This is the approach used in the *Bayesian Compositional Hierarchy* (BCH) [Neumann, 2008] which is a probabilistic scene model with the structure from an object-centric representation of a compositional hierarchy, similar to those used in formal logic ontologies (see Figure 3.3).

The hierarchical structure of the BCH allows strong mutual dependencies between objects to be confined to specific hierarchical groups of scene objects (or *aggregates*). Each aggregate is modeled by a joint probability distribution $P(AB_1 \dots B_K C)$ where A is a description of the aggregate, $B_1 \dots B_K$ are parts of the aggregate and C is the spatial layout of the parts. Each part B is itself modeled by an aggregate of lower level elements, unless it is the lowest level element in which case it is the result from an object detector (see Figure 3.4). For example, for a Balcony aggregate, A would be the bounding box for the whole balcony, B_1, B_2, B_3 would be parts with bounding boxes for Door, Window and Railing aggregates, and C would be the distance between the parts. The advantage of the BCH formulation is that joint probability distribution can be computed from each aggregate, represented as a multivariate Gaussian SVM and belief updates can be performed by propagation through the tree structure with a closed form solution.

In [Kreutzmann et al., 2009], the authors propose an incremental approach to performing detection of building facades using the BCH. When performing inference using this model, it is valuable to be able to consider multiple classifications of aggregates since a single low level misclassified element can propagate up and result in many misclassifications. Infer-

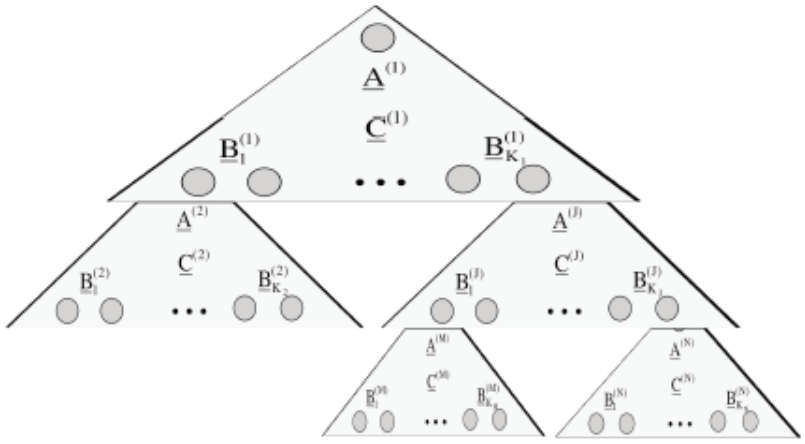


Figure 3.4: *The structure of a Bayesian Compositional Hierarchy. The triangles are aggregates defined by bounding box A, parts $B_1 \dots B_K$ and spatial layout of the parts C. Figure reproduced from [Terzić and Neumann, 2010]*

ence is performed using a technique called *beam search* which maintains multiple partial labellings of scene parts (which they call the "beam"). This approach maintains a set of partial solutions, ranked by their overall probability and adds labels with high probability, discarding models without any additional high probability labels.

Building facades are an interesting testbed for contextual object detection because of their visual and spatial properties. The components of the facade are very hard to classify using traditional sparse object detections. Structures such as windows, doors, balconies, vents, rails, etc, are all basically rectangular, homogeneously colored and have little texture. They do, however, have very specific spatial structure, dictated by factors such as the number of floors, of the building, the presence of the ground, the presence of a street, etc. They are perfect for testing an approach like the compositional hierarchy because there are many objects to detect and these objects are grouped together into larger structures.

We considered a probabilistic approach like the BCH for our model of object relationships but decided against it for a few reasons. While the scenes we are working with have some hierarchical structure (e.g., tables in offices and counters in kitchens) it is very limited and detecting enough objects to build up this structure is beyond the scope of our work. We were concerned that, without this hierarchical structure and the strong object-object influences within aggregates, their model would not be effective on the types of indoor scenes we are interested in. Also, [Kreutzmann et al., 2009] only uses very simple spatial relationships, not the broad range we wanted to try and we were concerned that the multivariate Gaussian BCH model would not fully exploit the large number of weakly interacting spatial

relationships in our work. If in our future work we handle larger numbers of objects per scene or across multiple rooms, then a hierarchical approach would be more useful.

3.4.4 Discriminative Models for Multi-class Object Layout

[Desai et al., 2009] provides the basis for our work in Chapters 7 and 8. Their goal is to use context, as defined by object-object co-occurrence and spatial relationships, to improve object detection rates. Their goal is to learn the difference between the types of spatial relationships found between true detection pairs and the types of relationships between true and false or false and false pairs. Rather than incorporate context into the object detection problem, they apply their model after object detection has been performed to remove false positive detections. A key difference between their approach and many others is that it is does not focused simply on the spatial relationships between the ground truth objects. Instead, they learn the informative spatial relationships for identifying the difference between true positive and false positive detections.

In this section we give only a brief overview of their model since it is described in more detail in Chapter 7 when we describe how we apply their model is 'to our work. The input to their model is $X = \{x_i : i = 1 \dots N\}$, a set of N object detections with an associated type and score computed from an image. Let $Y = \{y_i : i = 0 \dots N\}$ be a label vector which indicates if a detection is a true or false positive.

In [Desai et al., 2009] the stated goal is to determine the correct labeling of Y for a given X but they do not actually test this in their experiments. Instead they adjust the scores associated with all detections in X according to their 2D spatial relationships with all true positive detections in Y. This has the effect of decreasing false positive detection scores and increasing true positive scores. The likelihood of Y given an image X is evaluated using a scoring function

$$S(X, Y) = \sum_{i,j} w^T_{y_i,y_j} d_{ij} + \sum_i w^T_{y_i} x_i \tag{3.1}$$

where w_{y_i,y_j} is a pairwise weight vector that adjusts the score based on the likelihood that two objects of type y_i and y_j sharing the spatial relationship d_{ij} are true positives. $w_{y_i} x_1$ represents a local detector score associated with detection x_i for object type y_i. The spatial relationships d_{ij} that they use are fairly simple 2D qualitative spatial relationships and are illustrated in Figure 3.5. Inference is performed by computing $\arg\max_Y S(X, Y)$ to determine the best labeling of detections Y. They use a simple greedy search approach to compute $\arg\max_Y S(X, Y)$ and then adjust all detections in X using the weights w_{y_i,y_j} and w_{y_i} according to their spatial relationships relative to the true positive detections determined

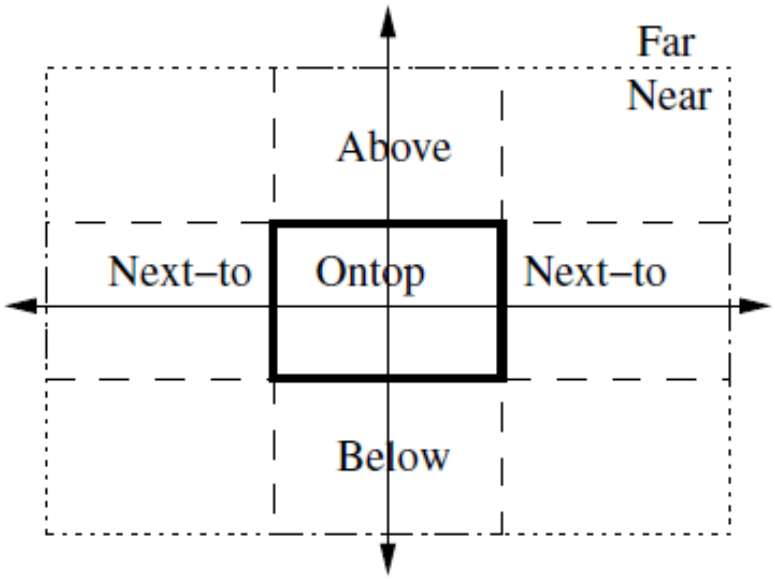


Figure 3.5: *The qualitative spatial relationships used by [Desai et al., 2009]. The relationships they used were distance-based (near,far), an orientation-based (above,below, next to) and a topological (on top). Figure reproduced from [Desai et al., 2009].*

by Y.

The model is trained with a set of images with true and false positive detections. The weights are learned such Y for each image closely matches the correct labeling of all detections as true or false. A more complete description of this method can be found in Section 7.6. A structured SVM is used to compute the weights. Desai *et al.* use the PASCAL challenge data set of 20,000 images, covering 20 categories of objects, for training and test. They show moderately successful results, with an overall improvement of 3% in average precision of the detections after applying their model, with larger improvements in individual classes. Some classes decrease in average precision but this likely just means that their model has not learned any effective relationships for improving detection accuracy.

Our work extends that of Desai *et al.* by moving to a 3D data set and 3D spatial relationships. We are provide a more careful and thorough examination and selection of the 3D spatial relationships used, drawing upon the field of qualitative spatial reasoning to identify good features for use on 3D scenes, as we discuss in Chapter 5. We also modified the inference and learning techniques used in [Desai et al., 2009] to use a branch and bound tree search which lead to significant improvements in average precision. Finally, in Chapter 8 we perform a number of experiments that examine how the properties of the detections used in training and test affect the overall results.

We believe our work actually better demonstrates the value of spatial relationships for improving object detection than [Desai et al., 2009] because of their choice of training data. The PASCAL challenge 2007, which provided the training and test data used in [Desai et al., 2009] encompasses 20 classes of object, covering a very wide variety of object types. This broad range of objects come from a correspondingly large number of types of scenes. As a result, co-occurrence of object types becomes a very effective way of improving detection rates. For example, if a car is detected in an image, it is unlikely that a chair will also occur and therefore all chair detections can have correspondingly decreased scores. From their work, it is unclear how much benefit they got from the spatial relationships. In our work, we restrict ourselves to training and performing detection in only the same type of scene so co-occurrence provides little information and instead we rely much more on spatial relationships.

Chapter 4

Scene Classification using Object Detections

Scene classification interests us because it allows a robot to select a spatial model to aid in object detection that is customized to a specific type of scene. Rooms are physical structures defined by geometric properties of the environment, usually walls and doors, while scenes are a semantic labeling of an area. A room may contain multiple scenes. For example, a open plan condo might contain a kitchen, dining room and living room in a single large open area. Scenes are defined by the objects they contain and the set of related tasks that occur within them [Southey and Little, 2006]. Scene classification should therefore be possible by detecting objects in the scene and using them to infer the scene type. The other major approach to scene classification has been to recognize scenes as a whole, identifying consistent visual elements across entire scene images, an approach that works well when scenes are visually distinctive. Techniques such as *Gist* [Oliva and Torralba, 2001; Torralba et al., 2003] capture global properties of an image and use them to classifying scenes. Our work predates many other more objects focused approaches to scene recognition, as we discuss further in Section 4.4.

In this chapter, we demonstrate that object detections, unaided by our qualitative spatial model, can be used to perform scene classification in images. We also show that combining detected objects with global properties of the image can further enhance performance. We present a novel method of scene classification that uses object detections to perform scene labeling. Object-based scene classification we believe is more effective for indoor scenes and generalizable to a large number of previously unseen indoor environments.

In our work, object occurrence information from the LabelMe database is used to inform classes of useful objects for detector training. We train these detectors automatically using the DPM object detection, discussed in Section 3.1.2. We then train an Alternating Boosted Decision Tree that uses detection scores to predict the scene type. We compare this method to using Gist alone on an indoor dataset. We then present another method that combines Gist as well as object detection scores, and show that the two types of cues, when combined, lead to better performance than when used alone. The work presented in this chapter was

jointly performed with Pooja Viswanathan [Viswanathan et al., 2011] and is included with her permission.

4.1 Scene Classification

Scene classification using semantic labels such as "kitchen", "bathroom", etc. and vision or shape-based methods has been an active area of research [Ersi and Tsotsos, 2012; Pronobis et al., 2010; Quattoni and Torralba, 2009; Wu et al., 2009]. However, most of these methods rely on the global properties of images. Some compute local image properties using feature-based methods but do not explicitly attempt object recognition. There have been significant improvements in object detection for robots [Meger et al., 2008] and the success of the embodied object recognition scenario presents the opportunity to leverage object detection for higher-level environment understanding.

We conducted early work that demonstrated that recognized objects and their locations can be used to automatically label scenes in the environment through the use of annotated databases, as demonstrated by the spatial-semantic modeling system [Viswanathan et al., 2009]. In this work we were able to label 2D regions of the map with the correct scene type using object positions in the map. This system, however, assumed the ability to recognize objects perfectly and did not address the recognition problem. Furthermore, the underlying classification mechanism was a simple probabilistic model which did not fully take advantage of available information provided by the object detections.

Labeling areas of a 2D map, such as that captured with Simultaneous Location and Mapping, with descriptive tags has most commonly been performed in topological mapping. In work by [Ranganathan and Dellaert, 2007] graph-like maps are constructed where nodes are classified using visual object recognition. Kröse *et al.* have developed a series of practical systems [Kröse et al., 2007; Spexard et al., 2006] in which the visual similarity between images is used to cluster regions in the environment. Scene labels for the clusters, however, are provided by a human through speech.

For classifying images of scenes, [Oliva and Torralba, 2001; Torralba et al., 2003] use global properties of a scene (Gist). [Pronobis et al., 2006, 2008, 2010] combine Composed Receptive Field Histograms, SIFT and laser data to perform scene classification in indoor environments, under different illumination conditions. Local regions are used to infer an intermediate "theme" of an image in [Li and Perona, 2005] to aid in scene classification. Several other context- and region-based approaches have been implemented, and can be found in [Bosch et al., 2007].

The authors of [Quattoni and Torralba, 2009] find that most methods that achieve state-of-the-art performance in classification of outdoor scenes perform significantly worse on indoor scenes. They observe that some indoor scenes are better described by the objects in them and thus combine global and local properties (Gist and spatial pyramid of visual words) to achieve increased performance. However, the reported multi-class average precision rates for the indoor dataset are still found to be low.

Object-based methods have also been used for scene classification, as in [Vasudevan and Siegwart, 2008], where functional regions of the environment are labeled based on object occurrences. However, the main drawback of this method was that it was trained on specific instances of objects and tested on the same objects under different viewpoints and lighting conditions. It remains a challenge to determine which objects are strong cues for scene classification. In addition, generic object class recognition has been a challenging task in computer vision research.

4.2 Automated Scene Labeling

We developed a system to categorize scenes based on object detectors learned from LabelMe images. Our system is composed of four components. Firstly, we perform automated data collection from LabelMe, thus facilitating the collection of training images used to recognize a large number of object categories. We compute a Count Model that represents the number of times an object is observed in each scene type in the LabelMe data based on user-provided text labels. We then use images from LabelMe to train windowed object detectors for the most frequently occurring objects. Finally, we use an Alternating Boosted Decision Tree (ADTree) to predict the most likely scene type given the detected objects in a scene. Furthermore, we show that enhanced performance can be achieved by incorporating global cues (such as Gist) into our framework.

4.2.1 LabelMe Data Collection

LabelMe [Russell et al., 2008] is an online database of user annotated images. In LabelMe, the user can annotate an object in an image by outlining a region of the image using a polygon and giving that region a label. The entire scene can also have a description contained in the filename. Figure 4.1 shows a kitchen scene from LabelMe with several labeled objects. We use LabelMe in two ways. The technique we use for training of object detectors requires tight bounding boxes that we can acquire using the LabelMe polygons.

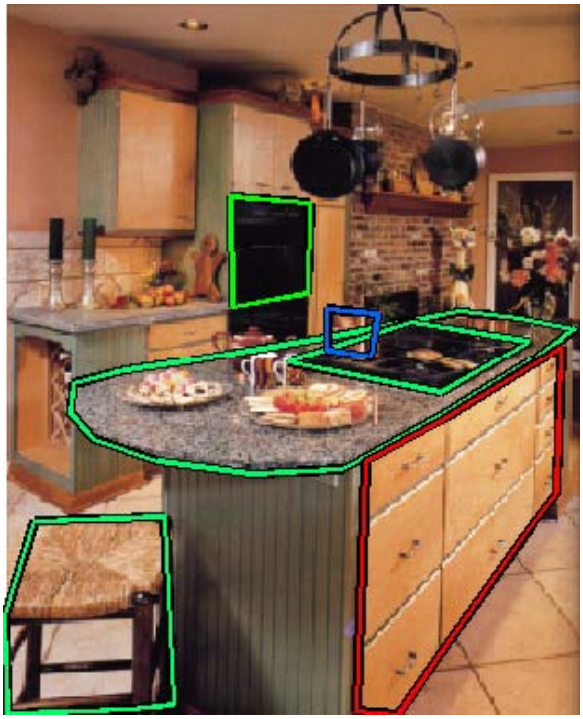


Figure 4.1: *A kitchen scene from the LabelMe database. The polygons used to segment objects in the scene are shown as colored lines.*

Also, our *Count Model* is computed using the correspondence between labels of objects in an image and the scene name descriptor found in the image filename. In creating this model, we do not directly analyze the images in the dataset, and instead focus on the textual annotations in each image.

4.2.2 Count Model

In order to perform scene classification based on objects, we first need to learn a model of the types of objects and number of occurrences in each scene type. We obtain this information from the LabelMe database by querying for scenes and recording the number of annotated occurrences of each object in the scene, as in [Vasudevan and Siegwart, 2008] and [Viswanathan et al., 2009].

The counts table $ct_p(o)$ contains the number of times object o occurs in images of scene type p. If the number of images of scene type p is n_p, the likelihood of observing object o in scene p is computed as

$$P(o|p) = \frac{ct_p(o)}{n_p} \tag{4.1}$$

We refer to this likelihood as the Count Model, which is used to inform detector training and learning of the Scene Model described below. We used only objects that appeared at least a 10 times in the scenes.

4.2.3 Useful Objects

The most useful objects for the scene labeling task are ones that provide the most amount of information gain. Information gain IG for an object o can be computed as in [Yang and Pedersen, 1997]:

$$IG(o) = -\sum_i P(p)logP(p) + P(o)\sum_i P(o|p)logP(o|p) + P(\neg o)\sum_i P(\neg o|p)logP(\neg o|p) \tag{4.2}$$

Upon analyzing the relationship between information gain and the frequency of occurrence of objects, we noticed a positive correlation between information gain and frequency as in [Yang and Pedersen, 1997]. The most informative objects are often the most frequently occurring, in the domain of scene labeling. This is because there is relatively little crossover between types of objects in scenes. In addition, since most LabelMe scenes contain objects in realistic home settings, objects that have high counts in the learned model are more likely to be present in the intended test environment (homes). Thus, it is sufficient to determine the most frequently occurring objects in each scene type to train object detectors. A histogram of these objects for some scene types can be found in the Section 4.3.

4.2.4 Detector Training

For object detection, we used the system created by [Felzenszwalb et al., 2008] discussed in Section 3.1.2. Their approach must be trained on images of the target objects with accurate bounding boxes, making many conventional data sources unusable. A database of labeled images is needed to train the object detector and in this earlier work we used the LabelMe database [Russell et al., 2008]: a free online data source which provides a large amount of human-labeled images, which contains indoor scenes suitable for scene labeling and object recognition. We trained object detectors for a subset of the most frequently occurring objects based on the object counts. A total number of 61 objects were used (corresponding to approximately 15 objects in each scene type). The precision-recall rates for a few categories, as well as visualizations of a detector model can be found in the

Experiments section.

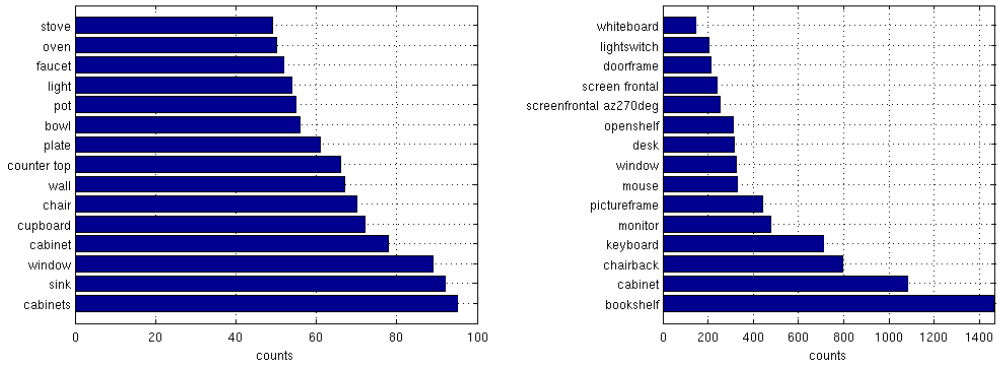
In our later work on improving object detections using 3D spatial relationships, discussed in Chapter 8, we stopped using LabelMe for providing training images for the object detector. We found use of user designated labels lead to poorly labeled images and multiple names for the same type of object. Also, there were insufficient examples of some types of objects and bounding polygons were inconsistently drawn. At the time we were working on the material presented here, ImageNet [Deng et al., 2009], which we use for our object detection in Chapter 8, did not provide bounding boxes for objects in images and was therefore not usable with the DPM object detector.

4.2.5 Scene Labeling Using Boosted Decision Trees

A decision tree is a hierarchical model for supervised learning that identifies local regions in the input space using a sequence of recursive splits which we described in Section 2.1.1. Significantly, boosted decision trees require no knowledge about the properties of the weak classifiers used and can be combined with any weak classifier that is more accurate than a random guess, allowing us to test a wide variety of features. Also, they are not prone to over-fitting and have no parameters to tune except for the number of rounds they trained and can usually just be trained until the test accuracy plateaus. To achieve multiclass classification, we trained a decision tree for each scene type using a 1 vs. all approach. We describe the inputs to the boosted decision trees in Section 4.3 since they vary between experiments.

4.3 Experiments

In this work, we attempt to classify kitchens, offices, bathrooms and bedrooms. However, due to the automated nature of our system, we could learn models for other types of scenes including living rooms and dining rooms by simply querying LabelMe for more scenes. We did not do so because there was an insufficient number of examples of these types of scenes in LabelMe to determine good Count Models and it is difficult to train object detectors for objects in these scenes from the relatively small number of examples in the LabelMe dataset.



(a) *Kitchens* **(b)** *Offices*

Figure 4.2: *Counts of the types of objects found in kitchen and office scenes.*

4.3.1 Count Model

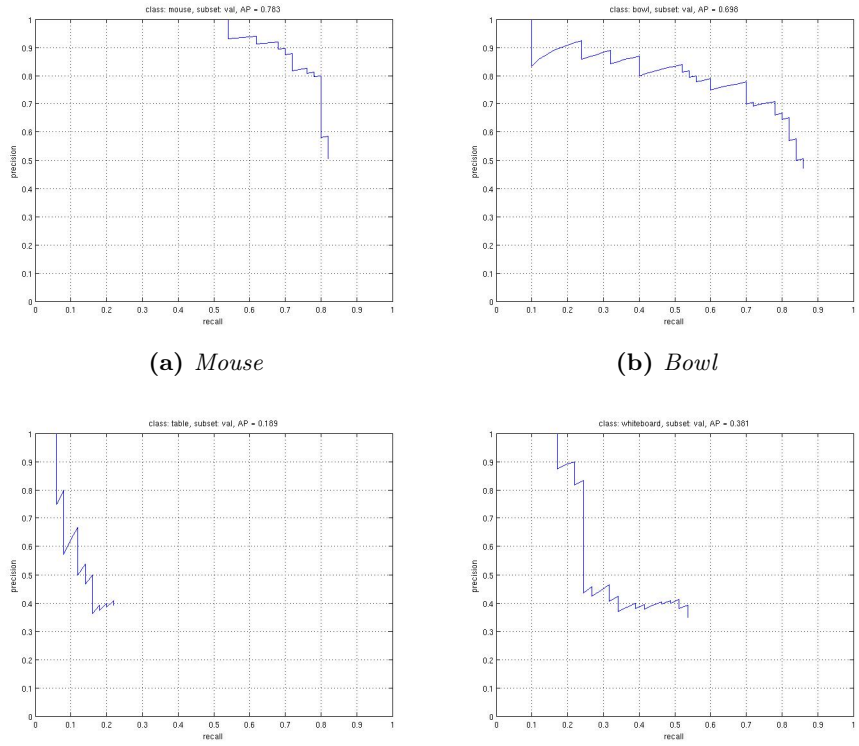
Figure 4.2 shows the object counts learned for kitchens and offices. We display the 15 most frequently occurring objects in each scene type. As seen in the figures, some of the objects have unusable labels due to the ambiguous user entries. We thus select a limited number of the objects, and show later on that these are in fact sufficient for the task of scene classification.

4.3.2 Detection

We trained object detectors using at most 200 positive examples and 1000 negative examples for each class. We set the number of components of the mixture model, n, based on the size of the training data for each class (classes with few training examples were trained on 1 component, while classes with more training data were trained on up to 3 components). Thus, training examples are split into n components based on the aspect ratio of the bounding boxes they contain. DPMs are trained on each component individually and merged together to form the final model.

In order to produce precision-recall and average precision rates for each category, we validated the models on images of LabelMe objects that were not used in training. We used loosely cropped versions of these images to prevent unannotated true positive examples in an image from being detected as false positives. Figure 4.3 shows some of the most and least successful detection results. As seen, objects that are usually partially obscured by other objects (furniture such as desks and tables) tend to perform the worst. Training images for

these classes mostly contain views of cluttered table/desk tops. Alternate views of furniture can be gathered by using other Internet sources, which would provide additional structure (e.g. table legs) for use in training detectors.



(a) *Mouse* **(b)** *Bowl*

(c) *Table* **(d)** *Whiteboard*

Figure 4.3: *The precision/recall rates of object detectors. Top rows shows 2 of the most successful classifiers and the bottom row shows 2 of the least successful classifiers.*

4.3.3 Scene Classification

We designed experiments to test scene classification in three different scenarios. In the first experiment, we classify scenes based on perfect labels of all annotated objects. In the second, we classify full images that depict a scene containing different types and numbers of objects, using real detection results.

Table 4.1: *Results for scene classification with perfect object labels.*

Scene	Prec.	Recall
Bathroom	0.96	0.88
Bedroom	0.92	0.92
Kitchen	0.95	0.93
Office	0.98	1.0
Average	0.95	0.93

Scene Classification with Perfect Labels

In this experiment, we want to determine the performance of the scene classification system using ADTrees if all objects can be recognized perfectly. We run 10-fold-cross validation on all examples of bedrooms (37), bathrooms (52), offices (647) and kitchens (190). The number of examples for each scene type is indicated in parentheses. Only objects that occur in at least 10 images are used as features in the ADTrees. In this experiment, inputs are binary, indicating the presence or absence of an object. As seen in Table 4.1, our scene classification algorithm produces assignments that closely match the ground-truth scene labels for all scene types. Bathrooms produce the lowest recall rate due to the limited number of example images currently in LabelMe and the sparsity of objects in these scenes. This demonstrates that objects present in a scene are very useful in classifying it.

Scene Classification with Object Detections

Our second experiment determines scene labels using real objects detections in the image. These images only contain a portion of the scene, and can contain few to many objects. The highest SVM detector scores, produced by running each learned object detector on the image, are fed as input to the BDT, which then infers the scene label. We found that multiple detections per object class resulted in negligible improvement, and thus used only the top detections.

We expanded the indoor dataset in [Quattoni and Torralba, 2009] with 50 more images for each scene type acquired from an online image photo collection to minimize overlap between the Torralba dataset (which uses LabelMe as a data source) and the images used in detector training. We compare our object-centric method with the technique in [Oliva and Torralba, 2001], which uses Gist for scene classification using an SVM. We perform 10-fold cross-validation on the entire dataset for both methods. Results of scene classification on the combined indoor dataset are shown in Table 4.2, in columns 1 and 2. Given the difficulty of the task, our model performs extremely well at distinguishing between the various scene

Table 4.2: *Classification results for scene classification using object detections, Gist and both combined on images acquired by humans.*

Scene	Object		Gist		Object+Gist	
	Prec.	Recall	Prec.	Recall	Prec.	Recall
Bathroom	0.56	0.58	0.63	0.56	**0.68**	**0.63**
Bedroom	**0.57**	0.53	0.53	0.53	**0.57**	**0.64**
Kitchen	0.61	0.62	0.53	0.57	**0.64**	**0.67**
Office	0.58	0.60	0.53	0.52	**0.66**	**0.60**
Average	0.58	0.58	0.56	0.55	**0.63**	**0.64**

types based on a relatively small set of trained object detectors. This demonstrates that object detections using DPM on our automated training method are reliable for the task of scene classification. We also notice that our object-centric method is comparable to Gist, outperforming it in most cases.

Finally, we show results of combining object detections and Gist in the third column. The input features in the BDT in this experiment are the highest SVM object detector scores as well as SVM output from Gist. We use the average precision and recall rates. We see that due to the complementary strengths of the methods, combining them results in enhanced performance for all scene types in the indoor dataset.

4.4 Influence on Future Work

This chapter demonstrated a system that can perform scene classification using object detections on an indoor dataset. Our object-centric method outperforms Gist on the indoor dataset. The combination of object detections and Gist leads to enhanced performance. We have shown that, with state-of-the-art object detectors trained with large, freely available data sources like LabelMe, we can effectively both detect and classify a wide variety of objects in realistic indoor images.

Further work on classifying scenes using object detections in both indoor and outdoor scene has been pursued by [Li et al., 2010]. Like us they combined object detections with global scene descriptors but included spatial information and many more types of object detections. They also included 2D spatial information between detections and achieved very impressive classification rates on 9 types of LabelMe scenes ($\sim$ 70%). Their work showed that this object-centric approach is applicable to a broad variety of scene types.

From the success in scene recognition from a single image, we can infer that it is possible

to determine scene type with good accuracy given multiple images from different angles. Given classification rates similar to our own or that demonstrated in [Li et al., 2010], a simple voting process over all images of the scene would give very high confidence given many scene images. Therefore, we assume in Chapter 7 that scene classification is possible and only apply a spatial model trained for the correct scene type.

Chapter 5

Qualitative Spatial Relationships

This chapter will discuss the use of qualitative spatial relationships for describing the relative spatial properties of objects in human environments. This chapter covers several key concepts relating to qualitative spatial relationships and their origins in qualitative spatial reasoning. It ends with an overview of the four major types of qualitative spatial relationships that are used at the basis for our experiments in Chapters 6 and 8.

5.1 Qualitative Spatial Reasoning

A *qualitative representation* is one which, as opposed to quantitative representation, "makes only as many distinctions as necessary to identify objects, events, situations, etc. in a given context" [Hernndez and Zimmermann, 1994]. Qualitative representations embrace the principle of parsimony, that a model should use the simplest representation necessary for a given task. Ideally, such representation would be based on human behavior, cognitive models and communication, since these provide insight into how humans simplify their mental representations of their environment. A representation should be both simplifying and function well in a wide variety of situations.

Much of the research presented here is drawn from *qualitative spatial reasoning* (QSR), a branch of reasoning that deals with finding qualitative descriptions of structures in space and then reasoning about the physical properties relative to each other. Qualitative representations are based on the application of a calculus of *predicates*, boolean valued functions that partition quantitative values. We are primarily interested in the qualitative spatial relationships used in qualitative spatial reasoning rather than the logical elements of QSR. *Qualitative spatial relationships* express continuous quantitative spatial reasoning using discrete symbols that simplify reasoning about spatial concepts [Cohn and Hazarika, 2001]. Coming from the field of logic, qualitative spatial relationships provide us with a complete and consistent approach to representing space.

The key problem in QSR lies in finding meaningful, relevant ways of quantifying continuous relationships so that they capture so called "common sense" information about the environments humans inhabit and are appropriate to a problem or situation. We believe that provided with a spatial model based on qualitative spatial relationships, a computer or a robot can interpret and conceptualize the physical world in a manner more closely resembling a human's and make predictions and actions better suited to its task.

Qualitative reasoning allows for inference in the absence of complete knowledge, not by treating the information in a probabilistic manner and modeling the uncertainty, but by purposefully grouping together like values that are conceptually similar. QSR allows a robot to consider its surrounding in terms of human concepts such as "near", "on top of", "inside", etc. While this information is contained in the quantitative spatial data, identifying and extracting it is part of the role of QSR. Allowing for uncertainty about spatial information simplifies many spatial problems since less accurate data is required. The shift from a continuous space to a discrete space decreases the complexity of spatial reasoning and learning by reducing the dimensionality of the problem, as well as changing the type of reasoning required.

Much of the body of QSR literature focuses only on 2D spatial representations. When dealing with indoor environments, 3D representations capture significant relationships between objects. The vertical placement of objects clearly has great significance indoors (e.g., consider the different treatment of food on the floor vs. food on a table). Advanced sensor devices like stereoscopic cameras and laser range finders can provide complex 3D information about scenes. Since 3D spatial relationships are not the norm in QSR, we have had to translate many traditional qualitative spatial concepts from 2D to 3D.

An important concept when discussing qualitative systems is the concept of *granularity*, the degree to which the components of a system are subdivided. A coarsely granular system is one comprised of fewer or larger components than a finely granular system. Granularity is often confused with scale, since a map that has a scale of 1:1000 has a coarser granularity than one that provides 1:100 scale. However, scale is only one dimension on which to consider granularity. Most real-world spaces can be considered at a broad range of granularities. An ideal qualitative spatial relationship should be as coarsely granular as possible while still containing sufficient information for all tasks within that environment.

5.2 Qualitative vs. Quantitative Relationships

Qualitative spatial relationships rely on decomposing and partitioning spatial relationships. Decomposing a quantitative representation of space breaks it down into multiple simplified representations based on some mathematical property of space such as topology, orientation or distance. These decomposed representations are then partitioned into discrete subsets that capture relevant divisions within that representation. Quantitative relationships can also decompose space to varying degrees but do not usually discretize it using partitions.

5.2.1 Advantages of Qualitative Spatial Techniques

The following is a list of advantages of using qualitative spatial techniques as opposed to quantitative ones:

Complexity reduction: An excess of information can often make learning tasks harder because of the time and difficulty involved in identifying relevant data. By translating spatial data from a continuous, combined multi-dimensional space to a discrete, decomposed space, the use of qualitative spatial relationships reduces the overall information needed to express the relevant properties of an environment [Escrig and Toledo, 1998]. Also, through decomposition, information of different types can be considered separately or in conjunction as necessary [Hernndez and Zimmermann, 1994].

Compensate for data inaccuracy: Getting exact quantitative data about the world is often difficult or expensive. The degree to which we are uncertain about the resulting information is often unknown. By using a representation that does not require exact information, we can potentially ignore or mitigate problems resulting from these inaccuracies.

Handling partial and uncertain information: Qualitative spatial approaches are able to handle vague or uncertain information about their environments by intentionally not differentiating between similar situations by using a more coarsely granular approach to identifying qualitative relations [Hernndez and Zimmermann, 1994].

Human inadequacy simulation: Our senses are not designed to precisely measure properties of space against abstract, non-present conceptual properties like "a centimeter' or "a gram" [Hernndez and Zimmermann, 1994]. Therefore, it seems reasonable that an approach to spatial understanding that aims to function in an environment created

by humans should base itself on a model similar to that used by a human. The suggestion that an approach is based on "how humans do it" is potentially risky without adequately exploring the actual mechanisms in the brain. However, humans and other animals are clearly good at reasoning in space without perfect information about spatial properties, which indicates that precise quantitative information is unnecessary [Renz, 2002].

Generalization: Reducing our reliance on exact values and combining like cases together makes seemingly different situations become similar. Approaches that learn rules from simplified examples are more likely to generalize to novel situations [Hernndez and Zimmermann, 1994].

5.2.2 Disadvantages of Qualitative Spatial Techniques

Qualitative spatial techniques have some potential drawbacks:

No universal approach: There are many qualitative spatial relationships that have been suggested in the QSR literature that rely on different spatial properties and are applicable to different tasks. Experimentation is often required to determine which approaches are best suited to a task and communication between different qualitative approaches first requires a definition of a agreed upon "spatial language" [Escrig and Toledo, 1998].

Varying granularity: Before attempting a task involving qualitative spatial reasoning, it is necessary to identify what granularity of information about the world is required [Escrig and Toledo, 1998]. If the level of granularity is not obvious, as is likely the case, multiple levels should be examined to approach an optimal solution.

Translational costs: Since most sensors acquire only quantitative information about the world, mechanisms must be in place to transform the data to a qualitative system. The costs for the conversion could be time, computation, or money (if, for example, a human expert is required to perform the translation).

Lack of Bi-Directionality: Moving from a quantitative representation to a qualitative one is typically a well defined and exact process since the shift reduces the overall information conveyed. However, moving back from a qualitative representation to a quantitative one is more problematic since previously exact values are now undefined [Hernndez and Zimmermann, 1994]. For example, a robot might know through QSR that a knife belongs on the table but to actually place it there it will need exact

coordinates in space that a qualitative model cannot provide. In our work, this is not a problem since we do not need to translate any qualitative relationships back into quantitative values.

5.3 Spaces

Before it is possible to discuss the spatial relationships between objects, we must consider the nature of the spaces involved. [Hernndez and Zimmermann, 1994] suggests that a cognitive model of space used for qualitative spatial relations should draw on the properties of four types of spaces:

Mathematical spaces that are comprised of mathematical elements interacting according to a set of axioms and which provide the tools that allow spatial concepts to be abstracted, formalized and applied. They provide the mechanisms for defining a mapping from the quantitative representation to a qualitative one. Euclidean spaces are an example of an often appropriate model and have the advantage of being an axiomatization of physical spaces[Hernndez and Zimmermann, 1994].

Physical Spaces where objects follow "real world" physical constraints [Renz, 2002] (often associated with Newtonian physics) such as:

- Physical objects are homogeneous, continuous and finite, which is clearly a useful simplification since object material is not always homogeneous.
- Objects have only positive extension (e.g., objects cannot have a negative width or height).
- Different objects cannot occupy the same space at the same time.
- A specific object exists only once in space.
- To move from one point to another, an object must pass through some space in between.

These constraints help restrict spatial models to situations that have a physical analog.

Psychological Spaces that define a perceived model of actual space. The properties of psychological spaces are based on physical spaces as filtered through a variety of human senses. Mental models of space do not seem to correspond to simple Euclidean spaces [Roberts and Suppes, 1967].

Metaphorical Spaces that apply when spatial concepts are applied to a non-spatial domain but where spatial concepts allow metaphorical insights. For example, the organization of a company might be described spatially, where influence and common goals are mapped onto position and overlap. The application of spatial rules outside the spatial domain allows for an analysis of the inherent properties of those rules[Hernndez and Zimmermann, 1994].

In Chapter 7, we discuss our approach for differentiating between true and false positive detections using spatial relationships. In this work, we are determining spatial relationships between detections which are an interesting combination of both a physical and psychological construct. Detections are based on physical objects and a true positive detection corresponds to an actual object in a scene and that object should obey all requirements we gave above for a physical object. However, a detection can also be considered as a psychological construct since it is based on a entity's perception of the environment. In our case the entity is a computer or a robot rather than a human. Psychological constructs do not need to obey the physical rules of an actual object since detections can overlap, have a non=finite extent, etc. So in considering the spatial relationships between detections, we have to mediate between these two representations and identify relationships that are appropriate to both spaces. For example, true positive detections should not significantly overlap each other, unless we allow for a part-based detection where object parts should overlap with the whole object.

5.3.1 Sizes of Space

[Montello, 1993] proposed that as the size of spaces increases, the reasoning techniques employed by humans operating in them become based on more coarsely granular representations. As humans are able to perceive less of a building, their ability to make precise judgments of quantitative values become less accurate and they shift to a more coarsely granular human mental model as environments becomes larger. To aid in discussion of qualitative relationships relative to environment size, Montello defined the following four classes of spaces of increasing size:

Figural spaces that are similar in size to the human body and that can be perceived entirely without moving (e.g., a table or a desk).

Vista spaces that are larger than the human body but which can be examined without movement (e.g., an office or a kitchen).

Environmental spaces that requires movement to be explored (e.g., a house, a mall, an office building).

Geological spaces massive spaces that are likely never completely explored (e.g. a town or a city).

Obviously, there are no hard boundaries between these types of spaces but if one examines the spatial reasoning techniques used, a shift in the granularity of personal representation happens as humans move from small to large spaces [Renz, 2002]. At the larger scales, travel mechanisms change the way people experience space and cognitive models of distance are more influenced by our perceptions. For example, a path in the park which you always walk through might seem much longer than the road that you drive, even though it is actually much shorter. Distance metrics need to be adjusted to encompass different scales of space. In environmental and geographic spaces it is often possible to simplify 3D space into 2D by ignoring height since it is orders of magnitude smaller than the other dimensions. QSR research is grounded largely in 2D representations because much of the origins of QSR is in spatial analysis at the map level where height can be safely ignored.

We are primarily interested in figural and vista [Swadzba and Wachsmuth, 2011] spaces since they describe the majority of human buildings. In vista and figural spaces, humans make relatively accurate estimations of the quantitative spatial properties of objects around them and a substantial amount of the environment can be observed without moving. The higher accuracy of human reasoning in small environments allows us to use qualitative spatial relationships that can rely on accurate judgments of space and knowledge of the objects in the environment. At these scales, many decisions about object placement are based on physical human properties such are arms' reach, human height, walking distances, etc., so physical properties of humans become a defining characteristic when determining the granularity of spatial relationships.

5.3.2 Spatial Object Structure

Typically, in mathematical spaces, the basic unit of space is the point and all other spatial constructs (such as lines and regions of space) are defined as sets of points. In QSR, the most commonly used basic unit of space is the spatial region [Vieu, 1997]. Regions define the space occupied by both physical objects and psychological spatial constructs like detections and it is between regions we determine spatial relationships. Regions are used rather than points because qualitative spatial relationships are predominately used to reason about physical objects and physical objects occupy volumes of space, not points. As [Simons,

1987] said "No one has ever perceived a point, or will ever do so, whereas people have perceived individuals of finite extent". Lower dimensional units like lines and points do not have physical manifestations and, if necessary, for describing mathematical properties, can be defined in terms of degenerate regions.

Given the almost infinite complexity of the region occupied by a real world object, most QSR techniques rely on simplifying transformations to produce a more coarsely granular description of an object's region using its boundaries. There are many quantitative systems for encoding the region such as triangle meshes, splines, and voxels. These are rarely used in QSR because of the complexity involved in determining relationships between representation regions. Commonly used approximations of object region include bounding boxes or convex hulls [Clementini and Felice, 1997].

When discussing the spatial properties of objects, it is important to differentiate between object shape and the region that defines a object in a space. An object's shape is independent of its position and orientation, while the region defined by an object is based on its position, orientation, shape and scale. It is common in QSR [Hernndez and Zimmermann, 1994] to distinguish between shape and scale. We examine the difference in greater detail in Section 5.4.4.

5.3.3 3D Spatial Representations as 2D

Often in qualitative spatial work 2D representations of 3D domains are used which simplifies finding qualitative spatial relationships. For tasks such as map analysis and image understanding, 2D representations are clearly appropriate. However, both maps and images are 2D projections of 3D environments and involve a loss of information. With city maps, if height is ignored there is usually minimal impact on the usefulness of qualitative relationships. For computer vision, the loss of information from using images can lead to more significant issues but is necessary due to the nature of cameras and the prevalence of data sets with only images.

An option we considered for our work but rejected was to use a *2D surface-based qualitative model*, a type of augmented 2D representation suggested by Hernandez [Hernndez and Zimmermann, 1994] that projects 3D spatial data onto parallel 2D layers. The intuition for surface-based models is that most human environments are comprised of flat surfaces and that gravity holds objects on these surfaces. The actual height of an object, therefore, is determined by object size and surface height. Since object relationships are heavily influenced by objects on the same surface [Hernndez and Zimmermann, 1994], the model

treats objects on the same surface differently than objects on different surfaces. With the surface-based approach, the space inside a building is treated as a collection of 2D layers corresponding to surfaces. Qualitative spatial relationships between objects are based on the object's 2D relationships on surfaces with an additional factor that a takes into account if the objects are on different surfaces.

In the end, we decided to focus our work on full 3D models of scenes because, with the increasing prevalence of 3D sensors, there will likely be more robots in the future using full 3D representations of their environments. If object detection can be performed a full 3D representation, then mapping to 2D surface-based one at the loss of some potentially valuable relationships is unnecessary.

5.4 Qualitative Spatial Relationships

Most qualitative spatial relationships are based on one of four of simplifying properties of quantitative spatial relationships between objects: distance, orientation, topology, shape [Cohn and Hazarika, 2001; Freeman, 1975]. The shape category here includes both shape and the related but separate property of scale. The four properties can be considered separately or in combination to develop systems for describing qualitative spatial relationships. Determining qualitative spatial relationships involves partitioning these properties into a set of exhaustive, pair-wise disjoint relationships. Since the type of qualitative relationships that are appropriate is dependent on the associated task, there exist many different ways of performing this partitioning.

Our experiments use primarily orientation and distance relationships but an examination of all four areas is necessary for later discussion of our work and is valuable to understand why we made our decisions. For each property used in our qualitative spatial relationships we describe the approaches for considering and quantifying these properties and the rationale and techniques used for creating the qualitative partitions .

5.4.1 Orientation

In Euclidean spaces, orientation is expressed using angles which define the direction of a vector relative to a universal axis. However, more generally, orientation is a triadic relation based on a target object T, a reference object R and a frame of reference FoR. A key difficulty in using orientation as a spatial relationship is identifying a frame of reference that is appropriate to the desired goal and that can be consistently identified in the environment.

Finding the right frame of reference is important because any orientation between R and T is possible given the choice of FoR (unless R and T are the same point, when orientation becomes meaningless).

There are three types of frames of reference used when determining orientation, each of which has significantly different interpretations:

Extrinsic frames of reference rely on a single external, invariant set of points as the frame of reference for all comparisons. Examples of extrinsic frames of reference include the traditional axis system used in Euclidean space, a fixed landmark such as the north pole or a vector like gravity. The difficulty with extrinsic frames of reference is identifying one that is meaningful for the space and task involved [Hernndez and Zimmermann, 1994].

For our work, the vector defined by gravitational direction is our main extrinsic frame of reference since it is easy to identify and gravity has a strong influence of scene structure. We also use a frame of reference defined by the walls of the room to determine x and y axes.

Intrinsic frames of reference rely on a physical property of R to define a frame of reference relative to T. To find an intrinsic frame of reference, object R must have recognizable features from its shape or appearance that make it possible to reliably identify enough points on R to create a frame of reference. Since it may be impossible to identify enough points on R to define a sufficiently high dimensional frame of reference, extrinsic orientation approaches should be able to gracefully change their granularity depending on the properties of R.

Since our early work described in Chapter 6 used synthetic data from a game, we had access to intrinsic frames of reference for each object, defined by its model in the game. Our early work used these intrinsic frames of reference but they were discarded once we realized that in our later real-world work we would not be estimating object pose.

Deictic orientation uses an embodied agent's point of view in the environment as the basis for defining a frame of reference [Cohn and Hazarika, 2001]. In many human interactions, a person is used as the reference point for comparative orientation, though using deictic frames of reference can be problematic unless both parties agree on which person is the reference (hence the age old question "Your left or mine?").

In our work, we have not found deictic frames of reference useful since there are no embodied agents other than the robot. Even though it is possible to use a robot's

sensors to define a deictic frame of reference, it is of little use in our work unless the robot had been involved in the creation of the environment and its position was known at the time. Deictic frames of reference for a robot are primarily useful when it needs to either understand or describe a spatial concept that is relative to itself (e.g., "Pick up the cup to your left").

An approach we believe would be useful in future work would be for a robot to "hallucinate" an embodied human in a scene where a human is most likely to be, then performing object detection using the resulting relationships. For example, the position of a chair in an office indicates that a human will frequently interact with the environment from that position. Spatial decisions have been made from that viewpoint, making it a useful basis for orientation judgments.

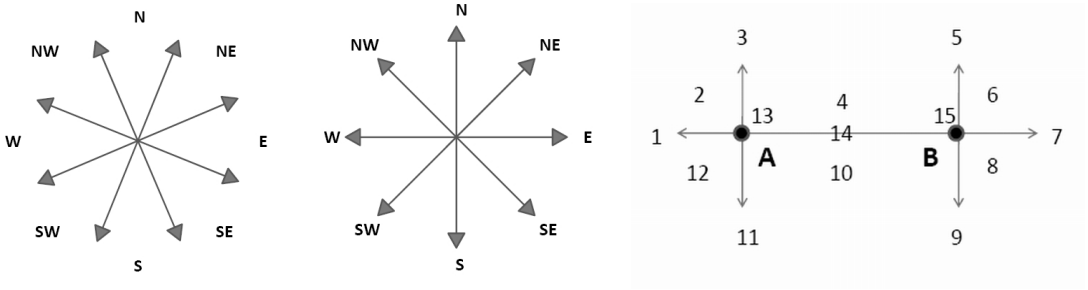
Orientation Between Points

Most approaches to qualitative orientation are based on comparisons between points (typically the centroids of objects T and R) relative to a frame of reference and are designed to work in 2D spaces [Hernndez and Zimmermann, 1994]. By judging orientation between centroids, these approaches are invariant to the shape of the objects and, if based on an frame of reference that is extrinsic to the objects, object orientation.

Possibly the simplest approach for describing orientation between points is that of [Schlieder, 1993] which, in 2D, expresses the relative orientations T and R given FoR as $+, 0, -$ for clockwise, collinear and anti-clockwise. Often coarsely granular approaches describe orientation using the cardinal directions. Instead of cardinal directions, "left", "right", "front", "back" descriptors are sometimes substituted when using intrinsic frames of reference. [Frank, 1991] identifies two basic methods for subdividing space for identifying cardinal orientations, shown in Figures (a) and (b) that are widely used and produce either 4 or 8 relations depending on granularity. Another widely used approach was developed by [Freksa, 1992] and is called the "double cross" calculus. It derives an intrinsic frame of reference from R. In 2D, there are 3 axes, one defined between R and T and two more orthogonal to that axis at R and T, see Figure (c), which divides space into 15 regions.

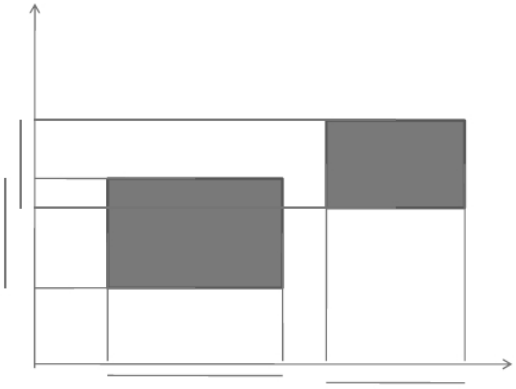
Orientation Between Regions in Qualitative Spatial Reasoning

Determining orientation relations based on object regions is a more complex problem than when using points because of the variety of possible object shapes[Hernndez and Zimmermann, 1994]. Region-based orientation approaches are not invariant to object position,

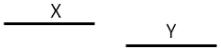
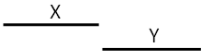
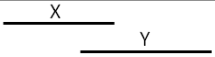
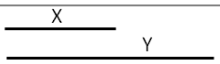
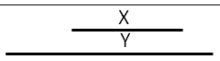
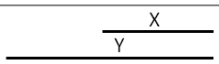
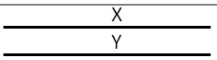


(a) *Conic Cardinal Dir.* **(b)** *Projection Cardinal Dir.* **(c)** *Double Cross*

Figure 5.1: *Three varieties of point based qualitative orientation systems.*



(a) *Axis Intervals*

Relation	Illustration
X **before** Y	
X **meets** Y	
X **overlaps** Y	
X **starts** Y	
X **during** Y	
X **finishes** Y	
X **equal to** Y	

(b) *Allen's Interval Algebra*

Figure 5.2: *A region based orientation system used to describe the projection of axis-aligned bounding boxes for the objects on to each axis of the frame of reference as shown in 5.2a. Intervals are compared according to Allen's interval algebra shown in 5.2b.*

orientation or shape but can capture more complex spatial relationships.

Most region-based qualitative orientation approaches rely on first reducing the shape to some consistent structure such as a convex hull [Barber et al., 1996] or an axis-aligned bounding box. Given an extrinsic frame of reference and a simplified structure, one common approach is to project the bounding boxes of R and T independently onto axes defined by a frame of reference (see Figure 5.2). The relationship between bounding box projections on an axis can then be defined using Allen's interval algebra [Allen, 1983], with one relation per axis of the FoR giving a total of 13^n relations for an n dimensional space. The number of

states becomes very large in more than a one dimension so often it is necessary to reduce the number of states in the interval calculus. Deciding which states to merge requires careful consideration of the problem and experimentation.

5.4.2 Distance

Distance is a scalar property based on the position and possibly shape and orientation of two objects. When dealing with distance measures, the first question is between what points should distance be measured? One solution is to determine distance between the region centroids of R and T, an approach that is invariant to object orientation and shape. However, using centroids results in an inaccurate distance if the distances involved are similar to the size of the objects and using centroids makes it impossible to identify some semantically meaningful (e.g., "touching", "overlapping", etc.). Otherwise, the shortest vector distance between the surfaces of the two regions can be used, or a reasonable approximation of that distance if the object shapes are highly complex.

When determining distance, there is also the question of whether to use the shortest path between the target and reference objects R and T regardless of the interposition of other objects or whether to find a path that passes through empty space. Determining the shortest free space path is a path planning problem and there are many known solutions that allow for additional variables such as minimum aperture sizes [LaValle, 2006]. We believe using the free space path instead of the shortest distance is only necessary when the two distances are significantly different and this mostly happens when comparing objects in multiple rooms or in very complex scenes. Another problem with free space distances is there needs to be a mechanism for handling situations where there is no path between the objects, such as when an the target or reference object is enclosed by another object.

Distance in Qualitative Spatial Reasoning

There are two general approaches to judging distance:

Absolute distance approaches use a consistent set of distance segments for all comparisons. Most absolute distance approaches are based on dividing the distance into sectors and assigning them a qualitative description such as "near" and "far". An issue with absolute distance approaches is that they do not scale across different sizes of space. For example, the meaning of a "near" object is different in vista spaces versus geographic space. When using multiple segments, a valuable technique for determining

their length is the *order of magnitude* calculus [Mavrovouniotis and Stephanopoulos, 1990] where comparison values are selected such that each sequential value is many times larger than the previous one. Using the order of magnitude calculus for describing distances has the desired effect that summing together multiple distances of the same qualitative length will usually result in the same overall distance (e.g., if A and B are near and B and C are near then A and C will likely be near). In our early work we used absolute distances based on the order of magnitude calculus.

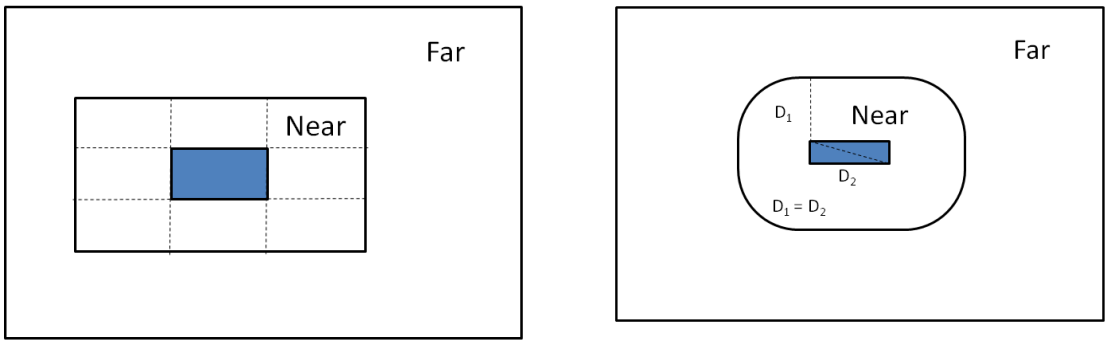
Later, we shifted to absolute distances based on *anthropometric* measures, distance measures based on average human physical capabilities. For example, the average human reach from a seated position is approximately *50 cm* while the average length of one step is approximately *1 m* [Tilley and Wilcox, 2002]. Objects are placed in environments so that they are comfortably available for the average person performing activities so distance measures based on simple movements can identify objects used for the same task or related tasks. Anthropometric partitions we can be selected to resemble an order of magnitude calculus and have most of the same advantages.

Relative distance approaches rely on a comparison to the size of other existing regions. For example, a simple relative distance metric used in QSR is the $CanConnect(X, Y, Z)$ metric [de Laguna, 1922] that is *true* if region X can join regions or points Y and Z. Relative distance approaches are often used in image based approaches to object recognition, thought they are not typically described as such. For example, [Desai et al., 2009] use a near/far qualitative spatial relationship between detection windows T and R where T is near R if $CanConnect(T, R, T)$ without changing the orientation of T as shows in Figure 5.3.

We became interested in relative distance approaches that use either the reference or target object as the basis for comparison, since we believed that they could scale between the figural and vista spaces in houses better and could even work in larger spaces. Relative distance relationships also do not require the determination of partitions in advance. They also can have the interesting effect of providing an asymmetrical distance relationship between objects such that smaller objects need to be closer other objects to be considered near to them.

5.4.3 Topology

Topology is a branch of mathematics belonging to the field of geometry that provides a coarsely granular approach to encoding object region structure through the examination of



(a) *No change in orientatoin* **(b)** *Any orientation*

Figure 5.3: *A example of a 2D relative distance relationship $CanConnect(X, Y, Y)$. The reference box Y is show in blue. The region surrounding Y shows the near/far partition. If X (not shown) overlaps with region surrounding Y, then X and Y are "near", otherwise they are "far". This first shows the partition region when there is no rotation of Y allowed to connect X and Y. The second shows the partition region when any rotation of Y is allowed to connect X and Y.*

connectedness. Connectedness can be considered as simply the property of two regions being in contact with each other. While QSR does borrow elements from mathematical topology, both point-set and algebraic topology are too abstract to be of practical use in QSR as they are focused primarily on representation and do not relate to the "common-sense" elements of human spatial reasoning [Cohn and Hazarika, 2001].

Topology in QSR is used as the basis for describing the overlap between pairs of regions X and Y. Cognitive studies indicate that topological relationships are highly important to human spatial cognition and have been shown to be the primary basis on which tasks such as grouping are performed [Renz, 2002]. Topology is also interesting since it is one of the simplest representation of space but still captures significant spatial distinctions.

The RCC-8 Topological Calculus

One commonly accepted approach to describing relative topology in QSR is the RCC-8 system (region connection calculus) [Randell et al., 1992] for defining the relative topologies of two bodies. RCC-8 describes an exhaustive, continuous pairwise disjoint set of topological relationships that can exist between two regions X and Y. Figure 5.4 shows a diagram of the following eight states of the calculus and the possible transformations between them:

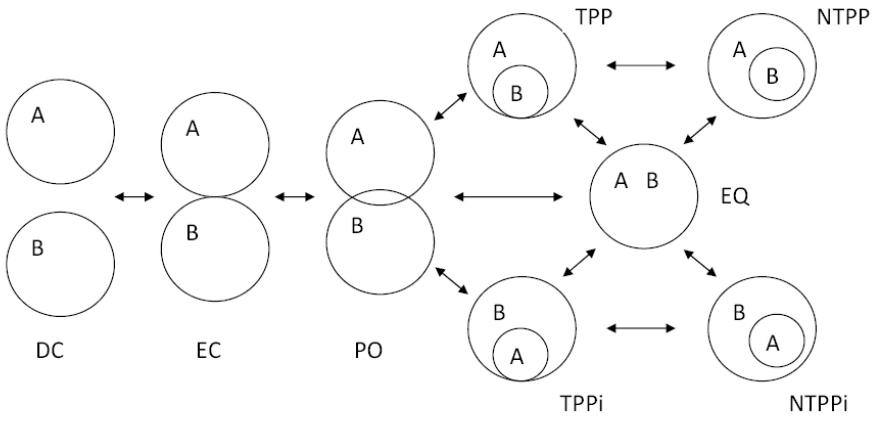


Figure 5.4: *The RCC-8 topological calculus.*

- *Disconnected (DC)*:
- *Externally Connected (EC)*:
- *Partially Overlapping (PO)*:
- *Equal (EQ)*:
- *Tangential Proper Part (TPP)*:
- *Tangential Proper Part inverse (TPPi)*:
- *Non-tangential Proper Part (NTPP)*:
- *Non-tangential Proper Part inverse (NTPPi)*:

All of the RCC-8 relationships can be defined in terms of a single primitive relation, the connected relation $C(a, b)$ [Renz, 2002]. A interpretation of $C(a, b)$ is that a and b are connected if and only if their topological closures share a common point. The closure of a region R consists of all points in R plus all the limit points of R. It can be shown that RCC-8 relationships can be applied to Euclidean spaces of any dimensionality [Renz, 2002].

In our work, we found topology a less useful that other factors such as distance for determining spatial relationships because in our work it is impossible for there to be overlap between physical objects, meaning that the only topological relationships between objects are contact relationships (e.g., "touching" or "separate"). However, we determine relationships between object detections and object detections can be considered as more of a psychological construct and can overlap each other. The localization error common to most object detections means that regions are almost never aligned such that they only overlap at their boundaries. This make many of the states in the RCC-8 topological relationships

(e.g., TPP, TPPi, EC and EQ) are extremely unlikely to occur.

5.4.4 Shape and Scale

The shape of an object can be defined as all the geometric information about a structure that is invariant to rotation, scale and position. Object shape provides a more finely granular approach than topology to describe the structure of individual objects. While topology can capture some aspects of structure such as the presence of holes, multiple parts or hollow interiors, it is insufficiently fine grained and expressive for some QSR applications.

Describing the shape of an object qualitatively has proven to be a difficult problem and it is not one we are actively pursuing since we are interested in exploring relative qualitative relationships between objects, not their individual qualitative attributes. Furthermore, the techniques we use for performing initial object detection in 2D in Chapters 7 and 8 prevent us from determining the actual shape of objects.

A separate but related concept is scale or size, which is a measure of the extent of an object in space and can be measured in a number of ways (e.g., axis interval, volume, area, internal span, etc.). A spatial relationship we were planning on using in our final experiments but eventually could not is size comparison. Comparisons between object sizes based on shape are invariant to object position and orientation. The size of an object can be measured in terms of volume, surface area, and internal spanning vectors, all of which are potentially meaningful. Being a scalar magnitude, the relative size of two objects T and R can be expressed based on binary magnitude comparisons $<, >, =$ [Hernndez and Zimmermann, 1994]. However, like many quantitative spatial properties, precise judgments of object size are difficult to make for humans. Therefore, when comparing relative sizes, it is useful to employ an order of magnitude comparison [Mavrovouniotis and Stephanopoulos, 1990] which equates object sizes if they are within an order of magnitude [Hernndez and Zimmermann, 1994; Mavrovouniotis and Stephanopoulos, 1989].

In our final experimental work in Chapters 7 and 8 we did not use any size comparison relationships because our technique for producing 3D detections gave all object detections of the same type the same size. Our triangulation techniques determined only the centroid of the objects, not the extent in 3D and all detections of the same type use a bounding box based on the average size of all objects in that class. Since all size comparisons would then equate to just comparisons of object type, the size relationships contained no useful information. However, we believe that size could be a useful relationship in future work if we had a technique for determining bounding box size from detections.

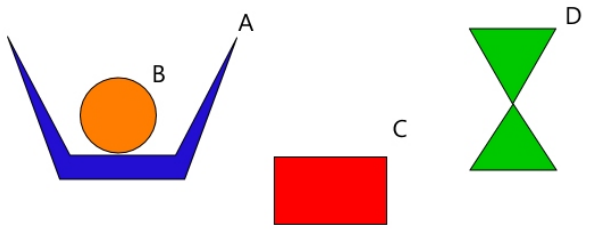
5.5 Proposed Complex Relationships

There were two additional types of relationships we included initially but which we decided not to use in our final work for several reasons. In this section we provide an overview of them for use in future work, explain why they were not included and how we replaced them with other relationships. We call the relationships *containment* and *betweenness*. These relationships relied on a mixture of representations and used a convex hull around the object regions to capture a perceptual extended region surrounding the objects used to make spatial judgments. For example, we would say an apple is "in" a bowl, even through there is no overlap between the actual bowl and apple. We consider a region to be "inside" the bowl, a region that can be approximated with a convex hull of the bowl. Both containment and betweenness relationships can be determined through examining the overlap of the convex hulls of regions defined by the target and reference object.

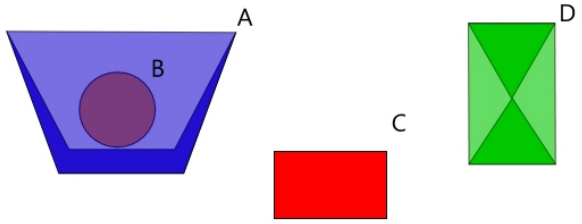
Containment

As mentioned in Section 5.4.3, we did not find the relative topology of object regions to be an effective property for finding containment relationships between physical objects because real world objects cannot overlap with each other. Instead, we were interested in a definition of containment able to recognize that a shelf could contain books or a bowl contain fruit, even though none of these objects physically overlap. In these examples, the object's containment relationships are based on properties of the regions of the involved objects. In the case of the shelf, a book is entirely within a recessed concave region in the shelf's structure. Similarly, the fruit in a bowl is either partially or completely inside a concave region of the bowl. What is required to identify these relationships is an approach that extends the regions defined by the containing objects and then a metric for examining the overlap between this extension and the reference object.

To identify containment relationships, we examine the interaction of the convex hulls of objects T and R [Aiello and van Benthem, 1999], which we refer to as $h(T)$ and $h(R)$. If the objects already use a bounding representation, like an axis-aligned bounding box, the convex hull does not need to be computed. Since now we are comparing the relative topology of two bodies that can actually overlap with each other, we can use the RCC-8 topological calculus which has been shown to be robust and effective at describing topological relations between regions. The result of our containment relation is an asymmetric relation $contain(T, R) = topology(h(T), h(R))$ that returns one of the 8 states of RCC-8 to describe the relative containment of a target object T and reference object R, as shown



(a) *Basic shapes*



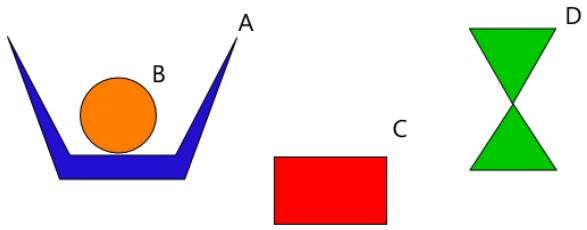
(b) *Individual convex hulls for identifying containment*

Figure 5.5: *This figure demonstrates how containment is determined using convex hulls. Figure 5.5a shows 4 objects defined by 2D regions. In Figure 5.5b, each shape has been overlaid by its convex hull to demonstrate containment detection. Object B is contained by object A, or more specifically, the RCC-8 containment relationship between A and B is NTPPi (non-tangential proper part inverse).*

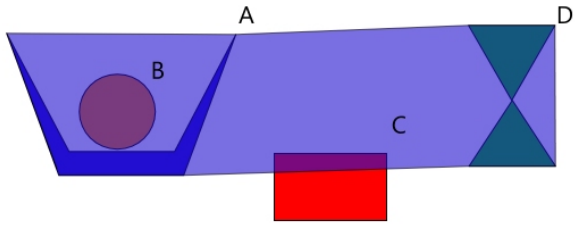
in Figure 5.5. The $contain(T, R)$ relationship captures the following types of containment relationships between R and T:

Containment Relationship of T to R	Equivalent RCC-8 Relationships of $contain(T, R)$
Separate	DC, EC
Overlapping	PO,EQ
Contains	TPP, $NTPP$
Contained by	$TPPi$, $NTPPi$

In the end we did not use the containment relationship for two reasons. Firstly, none of objects we included in our experiments in Chapter 8 are sufficiently concave that it is possible for other objects to be partially contained inside them. Secondly, objects inside containers are difficult to detect using cameras or 3D sensors. For many container objects, like refrigerators and ovens, objects inside can only be observed when the container is open. The few object that could open and contain other objects were never open during the data collection and did not contain any of the objects we were trying to detect. In an object detection model where a robot is observing and interacting with an environment for a longer



(a) *Basic shapes*



(b) *Shared convex hull (of A and D) for identifying betweenness*

Figure 5.6: *This figure demonstrates how betweenness is determined using convex hulls. Figure 5.6a shows 4 objects defined by 2D regions. In Figure 5.6b, a convex hull has been overlaid on the combined points of objects A and D to detect what objects share betweenness relationships with them. Object B and C are both between A and D, with object B having an NTTPi relationship and object C having a PO (partial overlap) relationship.*

period of time there would be more opportunities for it to observe opened containers and identify the objects they contain.

Betweenness

The spatial relationship *betweenness* identifies when a target object T is located in the interval between two reference objects R_1 and R_2. As a spatial relationship, betweenness can capture significant attributes of highly structured scenes [Aiello and van Benthem, 1999]. As a qualitative spatial relationship, betweenness may be of particular interest to us since some aspects of human environments are highly structured (e.g., books aligned on bookshelves or plates between knives and forks).

An obvious approach to determining betweenness uses the centroid points of the objects as values to a triadic, extrinsic orientation metric. Let X, Y and Z to refer to the centroids of objects R_1, T and R_2 respectively. In the simplest representation, if Z lies on the line between X and Y, then $\mathit{betweenness}(X, Y, Z) = \mathit{true}$. Since perfect collinearity may not match the desired definition of betweenness, the orientation typically should be relaxed as

follows. If $\varphi(X, Y)$ is the angle between X and Y given Z and α is a constant minimum angle then:

$$Between(X, Y, Z) = \begin{cases} true & \text{if } |\varphi(X, Y)| \leq \alpha \\ false & \text{else} \end{cases} \tag{5.1}$$

A problem with the orientation approach to betweenness is that it relies on object centroids but does not consider the object shape or size, potentially rejecting instances where part of T is between R_1 and R_2 but not the central mass. Also, the orientation approach requires an arbitrary constant α. Orientation as given above restricts the possible qualitative betweenness relationships to a binary value and fails to differentiate between an object partially between or completely between two other objects.

The *extended convex hull* approach to determining shape-based betweenness [Aiello and van Benthem, 1999] avoids the previous problems. In this approach, an extended convex hull surrounding both R_1 and R_2 is determined and compared topologically with T. Figure 5.5 shows an example of how the convex shape-based approach can be applied to determining betweenness in 2D. To determine shape-based betweenness, a convex hull Q is found for all points on both R_1 and R_2. The output value of relationship $between(R_1, R_2, T)$ is the RCC-8 calculus value of the topological relationship between Q and T. As shown in Figure 5.6, the convex hull approach to betweenness incorporates object shape, allows for a more complex set of possible potential relationships than the triadic orientation approach and avoids the α constant.

A problem with the extended convex hull approach is that anything inside a reference object would also be considered between the reference objects. To deal with this, we suggest that Q can be defined as the complement of the convex hull Q and the hulls surrounding the reference objects R_1 and R_2 individually:

$$Q = S(R_1 \wedge R_2) - (S(R_1) \cup S(R_2)) \tag{5.2}$$

where $S(X)$ is the convex hull of region X.

We did not use the betweenness relation in our work in Chapter 7 because it is not a binary spatial relationship and, given our limited amount of training data, we did not have sufficient examples to learn from. In effect, it was simply too finely granular for our work. To avoid this issue, we created binary relationships which captured similar spatial arrangements. One relationship is "beside' which compared the interval overlap of pairs of objects in the x and y axes defined by the shape of the room. In effect, this approach replaces the second

reference object which provided the frame of reference with the shape of the room providing the FoR. It is based on the observation that objects are aligned with the walls of a room. We also added a relationship called "vertical orientation" which identifies when objects are vertically stacked. Both of these relationships are explained in greater detail is Section 7.7.

Chapter 6

Spatial Object Classification in Virtual Environments

This chapter describes our early work on object classification using spatial relationships. The goal of this work was to test whether the spatial relationships between a target unknown object and a surrounding set of known objects could be used to effectively classify the unknown object. Note that this is a classification problem, not a detection one; we are assuming knowledge of the position of the target object and the position and type of all the reference objects. If an object can be identified by other surrounding objects, this would imply that object-object 3D spatial relationships could provide a good basis for performing joint object detection, which we discuss in Chapter 7.

6.1 Problem Formulation

The classification task was to identify a target object T given its spatial relationships with all the other reference objects $R_{1...N}$ in an entire house. The types of all objects except for T were known. In this work we did not have information on the shape of the building itself, only the objects inside it, so we had no way of determining the extent of rooms. Therefore, the spaces we were operating within were entire buildings, not rooms or scenes. The spatial relationships were acquired using information about the position, orientation and shape of T and $R_{1...N}$. This classification task, then, equates to a robot classifying a single object with perfect information about its size and shape but no evidence to its type given perfect information about all other objects in the house. Clearly, this is an unrealistic problem formulation but we picked this problem as the performance would provide an upper bound on how well object-object spatial relationships alone could be used for object classification. We use a maximum entropy model and alternating boosted decision trees for the classification task but in this chapter we will only cover the more successful decision tree results that outperformed the maximum entropy models in every experiment.

6.2 Synthetic Data from Elder Scroll Oblivion

In order to create and test this qualitative spatial relationship-based object classifier, we needed a data source with information about the type, shape, position and orientation of all the objects in a set of indoor environments. Acquiring this data through manual measurement was impractical for such early work. We were also concerned about the privacy issues involved in modeling the interiors of people's houses. At the time, using 3D measurement devices such as laser range finders and stereo cameras was deemed too problematic given the difficulty in building up complete 3D models of the environment, segmenting these models into objects and accurately labeling these objects.

For training and test data for our work, we identified a novel source of high quality synthetic data from a commercial video game called Elder Scrolls 3: Oblivion [Bethesda Softworks, 2006]. Figures 6.1, 6.2 and 6.3 show scenes from Oblivion: a tavern, a dining room and a library. The spatial relationships demonstrated in these scenes show a surprising level of complexity and detail. For example, the wine rack in the back of the tavern contains many different types of wine placed into its lattice structure. The dining room table is set with over 15 types of food, with different arrangements of food on the dinner table and serving plates. Other complex relationships include tools standing upright in barrels, bowls piled with fruits, and books ordered into sets on bookshelves. These models represent a snapshot of the contents of a house, so there is no temporal aspect to the data they provide.

6.2.1 Advantages of Using the Oblivion Data Set

A number of factors made the data from Oblivion appropriate for our early work. First of all, there is plenty of material. The designers of Oblivion also designed their game to be easy to modify and customize, so the data files could be parsed to extract the layout of the entire game world and there are tools for extracting the object models. Oblivion contains several hundred houses, castles, forts, churches, mines and other types of habitations, though in our work we restricted ourselves to places containing the word "house" in their description.

Houses appear in different architectural and ethnic styles ranging from city to city and none of the houses are simply identical copies of each other. The contents of each house are modeled at a very fine level of detail. Tables in the houses are set with meals, shelves are full of books and ornaments, bedrooms are laid out with beds, dressers and clothes. Rooms vary in the degree of tidiness with some carefully and ornately laid out and other messy or even derelict.

Figure 6.1: *A tavern from Oblivion. The wine rack in the rear of the bar holds about thirty bottles of wine in five different varieties. These were collapsed into a single object type "wineBottle" for classification purposes. The cat-like figure behind the bar is the owner. All game characters were removed from the training and test data.*

Altogether, there are over 1000 different types of objects modeled for the game. The setting is a medieval fantasy world so the contents of the houses were antiquated but the complexity of their environments is substantially beyond what was normal in any other game at the time.

Oblivion is an unusual game since almost all of the objects contained have no purpose but to make the environments seem realistic. In fact, many of the houses we used would never even be seen from the inside by the player, existing primarily as an interesting backdrop. This emphasis on realism rather that playability is a very important factor in using video game data since in most games the realism of an area is reduced to meet the artificial requirements of game play. For example, in many games, if the player were just supposed to move from one end of a building to the other quickly, most of the rooms in the building would not be modeled. Even if a room is modeled, most of the contents will be *game artifacts*, objects that are primarily significant to the game play, like weapons or "power ups" that would be artificially placed and not appear in the real world. There would only be a small number of background objects to make the environments realistic.

Figure 6.2: *A large and ornate dining room from Oblivion. In the center of the image is a dining table containing food and wine set for ten people. Surrounding the table are chairs, though these are partially obscured by shapes that show how character models would transition from standing to sitting on the chair.*

Figure 6.3: *A library from the Oblivion data set. The shelves in the back contain books, ornaments and tools. In the foreground there is a table set for two with food and wine.*

6.2.2 Issues With Using the Oblivion Data Set

There were also some fairly serious limitations of the Oblivion data set. Due to a feature of the game's design, none of the completely enclosing containers (e.g., boxes, chests, or drawers) in Oblivion actually contain objects but bookshelves, not being enclosed, do. Also, despite the high level of detail, there are fewer objects in an Oblivion house than in a real one. Since this was early work on this problem, we decided it was appropriate to use a simplified data set. Despite the medieval setting, the objects still have a spatial structure that can be modeled and the design of the houses is relatively modern, with many having multiple floors, large rooms and windows and areas like bedrooms, kitchens, and offices.

The bigger problem we had was convincing people that the game's environments were realistic enough to prove that our model had validity in the real world. The main issue that was raised was the possibility that some of the scenes might have been algorithmically generated by a computer with a simple set of spatial relationship rules. If this were true, it would imply that we were really only reverse engineering the rules used to create the environments in the first place. We saw no evidence of this when examining the data but it remained a concern and was part of the reason we stopped using this data in our later work.

6.3 Qualitative Spatial Relationships

The structure of a qualitative spatial relationship between a target unknown object T and a known reference object R was expressed as TSR where S is the spatial relationship (e.g., cup leftOf wine). Since this was early work, the relationships we used were ones of our own design, rather than specific relationships from the existing body of qualitative spatial reasoning. We used a very simple decomposition of space proposed by Freeman [Freeman, 1975] that has three types of relationships: distance-based (near, far), direction-based (left of, right of, in front, behind, above, below) and containment-based (inside, outside, surrounding, overlapping).

6.3.1 Distance Relationships

The distance relationships we used were: *touching*, *near* (approximately within arms reach), *mid* (short walking distance), and *far* (long walking distance). Our distance metric used the minimum distance $D(T, R)$ between the exteriors of the triangle meshes describing

the regions of object T and of object R [Gottschalk et al., 1996]. Each relationship then corresponded to a range of values along this distance. The range values we used were based on estimations of the conceptually relevant distances.

Touching: Physical contact between objects (*touching*) is usually caused by gravity and one object supporting the other. Objects typically only touch a small range of other objects so contact can be a highly descriptive relationship (e.g., food on plates, books on bookshelves and furniture on floors). It was formally defined as

$$T\,touching\,C \iff D(T,R) <= 0 \text{ cm}$$

Near: Objects that share a common purpose, such as eating, cooking or writing, are frequently found within arms' reach of each other and the *near* relationship can capture this effect. This range is particularly useful for grouping together objects at the tabletop level. It was formally defined as

$$T\,near\,C \iff D(T,R) > 0cm, D(T,C) <= 40 \text{ cm}$$

Mid: The *mid* proximity range captures objects that belong to a common task and share either a room or a scene. This range is useful for grouping together larger objects, like chairs and tables, at the room level. It was formally defined as

$$T\,mid\,C \iff D(T,R) > 40cm, D(T,C) <= 200 \text{ cm}$$

Far: This final proximity range relates objects that share the same overall environment (i.e., the same house) but are most likely not functionally associated except at the broadest level. This relationship is useful for extracting data on the overall object population of a house. It was formally defined as

$$T\,far\,C \iff D(T,R) > 200 \text{ cm}$$

6.3.2 Direction/Containment Spatial Relationships

Since early on we believed that our simple direction and containment relationships were not sufficiently expressive to be useful individually, we aggregated them together to form a joint direction/containment relationship. To capture directional and containment-based spatial relationships, we compared axis-aligned bounding boxes around our target object

T and comparison object R. For our containment relationships we used a classic simple qualitative set of relationships: *inside*, *overlap*, *surround*, and *disjoint* [Freeman, 1975].

The *disjoint* relationship was then subdivided further using directional data. We used an approach shown in 5.4.1 that compared the overlap of the object bounding boxes after they were projected onto the vertical axis using a simplified interval calculus. This subdivided the disjoint containment set into *above*, *below* and *level*. We opted not to use intrinsic qualitative spatial relationships (e.g., leftOf, behind, etc) because, to do so in any real world future work, we would need to reliably be able to determine the orientation of reference object R if it were to function as a frame of reference for T .

For complete set of relationships, let t be all points in the bounding box of T and r be all points in the bounding box of R. Let $min_{x,y,z}(T)$ and $max_{x,y,z}(T)$ be the the minimum and maximum values in each axis of the bounding boxes for object T.

$$
\begin{aligned}
T inside R &\iff t \subset r \\
T surround R &\iff t \supset r \\
T overlap R &\iff t \cap r \neq \emptyset, t \not\subset r, t \not\supset r \\
T above R &\iff t \cap r = \emptyset, min_z(T) > max_z(R) \\
T below R &\iff t \cap r = \emptyset, max_z(T) < min_z(R) \\
T level R &\iff t \cap r = \emptyset, \text{else}
\end{aligned}
$$

6.4 Relationship Sets

With the simple relationship types described above, we wanted to know how useful they were for classification and how they could be combined to improve classification accuracy. To that end, using our distance relationships and direction & containment relationships, we derived four relationship sets for our experiments:

Distance relationship set(*Dist*) This set used just the distance relationship.

Direction/Containment relationship set (*DirCon*) This set used just the direction & containment relationship.

Aggregate relationship set (*Aggregate*) This relationship set concatenates the *Dist* and *DirCon* sets (e.g., T near R, T above R).

Paired relationship set (*Paired*) This relationship set used the Cartesian product of

the two sets, merging relationships to create new a single relationship that combined distance, direction and containment (e.g., T near&above R). Since we are assuming that people use a common mental representation as the basis for their proximity, direction and containment judgments, the use of paired relationships are justified [Duckham et al., 2006]. The paired relationship set was interesting since it directly encoded more complex and specific spatial relationships than the aggregate set but at the cost of being more finely granular and not allowing the classifiers to combine relationships on their own.

6.5 Experiments

With our existing classifier-based alternating boosted decision trees, we ran some experiments to examine the effectiveness of different groups of qualitative spatial relationships for spatial object classification. Our goals with these early experiments were to determine whether qualitative spatial classification was a tractable problem and which spatial relationships were useful.

6.5.1 Training and Test Data

Our training and test data consisted of 197 houses from Oblivion containing approximately 14,000 objects with class names, 3D shapes, positions and orientations. The shape of each object is from the game's physics engine, which provides simple tight bounding polyhedra. We removed a number of game artifacts and atmospheric objects like cobwebs, ambient light sources, and hidden switches since they would either have not been placed by humans or were relevant only to the game. Objects were grouped together into classes like book, food, table, and chair, to produce 97 object classes and from these we selected 17 types of objects to classify. These 17 types were chosen because they appeared frequently, though with radically different frequency from each other, and they cover a variety of tasks and locations in the building.

To encode this data for the decision trees we use a flat, fixed length feature vector. If Z is the number of types of relationships used in the experiment and C the number of classes of objects, the feature vector has $Z \times C$ categories. Each feature then describes the relationship between the target object and a type of reference object (e.g., target near book). We include an "absent" relationship for each category to handle situations when there is no reference object of the category's type in the building. If there are multiple occurrences of the same

Table 6.1: *Relationship set accuracy comparison*

relationship Set	ADTree Accuracy
Direction/Containment	57.56%
Distance	59.97%
Aggregate	68.07%
Paired	66.09%

.

Table 6.2: *Confusion matrix for aggregate distance & direction/containment classifier*

		Classified Object Type (max values in **bold**)																
		Plate	Cup	Pitcher	Shelf	Vase	Beer	Bowl	Bed	Bench	Winerack	Book	Table	Candle	Chair	Painting	Food	Wine
Correct Object type	Plate	**342**	26	16	2	8	0	21	3	0	1	4	15	3	9	0	183	11
	Cup	16	**339**	21	0	5	12	5	1	1	0	0	2	6	9	0	76	12
	Pitcher	10	51	**130**	3	6	4	18	5	0	0	1	4	7	30	3	42	14
	Shelf	3	4	2	**141**	1	1	3	1	5	2	0	32	0	23	3	11	0
	Vase	10	21	24	1	12	2	6	1	1	0	0	2	7	10	0	**20**	3
	Beer	1	16	1	0	0	**45**	2	2	0	2	1	1	0	3	3	12	11
	Bowl	44	16	20	1	1	2	**78**	4	0	0	0	6	2	10	2	58	8
	Bed	1	2	7	5	1	0	1	**85**	2	0	0	14	2	12	2	7	3
	Bench	3	0	0	2	1	0	0	0	**35**	0	0	22	2	21	1	4	2
	Winerack	1	1	0	1	0	1	0	0	0	**20**	0	3	0	0	0	0	9
	Book	3	3	5	0	1	0	3	0	0	0	**9**	8	0	4	0	7	1
	Table	23	3	3	32	7	0	5	9	9	1	1	**307**	4	40	4	13	13
	Candle	7	14	12	1	4	1	3	0	2	0	1	7	**11**	21	1	13	5
	Chair	6	5	18	3	2	0	4	10	3	2	0	21	10	**459**	0	28	12
	Painting	1	0	2	8	1	0	0	4	0	0	0	2	0	1	**48**	3	2
	Food	78	43	22	3	1	1	13	2	0	0	0	15	2	20	2	**1650**	12
	Wine	4	8	4	0	0	4	1	1	1	1	0	1	2	13	2	43	**485**

type of reference object in the building, we used the closest reference object to the target object, under the assumption that closer objects are more significant for classification.

6.5.2 Results and Discussion

Table 6.1 shows the results of our experiments that compared each of the relationship sets classification ability. We found our alternating decision tree (ADTree) classifier was substantially more accurate than the maximum entropy model we tried with all the data sets and showed the greatest improvement on the sets that combined distance, direction and containment.

The direction/containment relationship set was the least useful relationship set, most likely because it did not differentiate between near and distant objects if they were disjoint. Distance relationships made for a better classifier indicating the significance of touch relationships and the importance of being able to differentiate between close and distant objects. The relationship sets that used both direction/containment relationships and distance re-

lationships produced a more effective classifier, showing that both are useful at different classification problems.

Paired relationships were less successful at fusing the distance and direction/containment relationships since, although allowing for complex dependencies on spatial relationships between the objects such as "above and touching", the decision trees could still recognize those complex relationships by encoding them into the tree structure. To recognize "above and touching" with only the component relationships requires two connected sequential decision nodes that are predicated on "above" and on "touching". Using the paired relationships limited the types of decision structures the tree could exploit but the differences were not large. Overall, what was evident was that having many spatial relationships was more important than how they were combined.

Table 6.2 shows the confusion matrix from the aggregate relationship set classifier, with the maximum values for each row in bold. Examining it shows that classified objects were often confused with types of objects found in similar locations. For example, benches were mistaken for chairs and bowls were mistaken for cups or plates. Overall there is an issue with objects being misclassified as food. Since food is a by far the most common object in Elder Scrolls 4 and found in a wide variety of places, objects that are difficult to classify were identified as food.

6.6 Influence on Later Work

These results showed us that the immediate scene context, as described by qualitative spatial relationships between objects, can provide significant evidence towards an objects type. It also showed us the representation of the spatial relationships (combined together or independently represented) did not matter as much as having a variety of relationships. The next step for our work was to solve the more realistic problem of combining spatial relationships with the results of a visual object detector to perform object detection using both visual and spatial information, the problem we address in the next chapter. We also knew that we would need to expand our selection of spatial relationships.

This was also the last time we used our synthetic data from the game Oblivion. Performing visual object detection on the video game objects was problematic because all examples of the same type of objects look exactly the same. We had proposed simulating the results of an object detector but decided that simulating a detector on simulated data would not to produce convincing results. The formatting of the Oblivion data was also difficult to work with and designing a system which allowed for switching between real and video game

data would have been very time consuming. Therefore, we dropped the Oblivion data and focused on collecting real scene data, as we discuss in Section 8.1.

Chapter 7

Improving Object Detection using 3D Spatial Relationships

In this chapter, we describe our approach to improving object detection accuracy using 3D spatial relationships. Given that object detectors typically produce many false positive detections, we want to differentiate between true and false positive detections using 3D spatial relationships. Our approach employs a model, trained on 3D examples of indoor scenes, to improve the accuracy of multiple types of object detectors applied to a scene, based on the spatial relationships between the detections.

We begin by describing our problem and the 3D object detections that are the input to our approach. The model we employ, which is based on the work of [Desai et al., 2009], was discussed in Section 3.4.4. In this chapter we describe how we adapt their model to 3D object detection, our improvements to training and inference using branch and bound tree search and how the parameters to our model are computed using structured support vector machines [Tsochantaridis et al., 2004]. We end the chapter with a detailed description of the 3D spatial relationships we employ.

7.1 Overview

Our work in Chapter 6 described a method for classifying a single object given perfect information about the type and location of surrounding objects. Our new approach, however, is based on object detections and the 3D spatial relationships between them, information that would be available to a robot. Our work is based on a technique for improving the object detection accuracy in images by [Desai et al., 2009] which they describe as a method that "simultaneously predicts a set of detections for multiple objects from multiple classes over an entire image". Our work performs a similar function but is applied to 3D detections and uses 3D spatial relationships.

The input to our model is a set of hypothesis boxes. In our work we use the term *hypothesis boxes* instead of 3D detections. A hypothesis box is a region of 3D space identified as containing an object. We differentiate between the terms because hypothesis boxes could be produced directly by an object detector (either image or shape-based) or constructed from image-based object detections. The output of our model is a new set of confidence scores for the hypothesis boxes such that the scores for true positives increase and false positives decrease. Our approach does not produce new detections, so it cannot improve on missed detections.

The model uses a structured SVM [Tsochantaridis et al., 2004] to learn a set of weights, each associated with a target object type, a reference object type and a qualitative spatial relationship (e.g., pan ABOVE stove). These weights capture whether the presence of a spatial relationship between two detections of the appropriate types is a good indicator that both detections are likely true positives. The weights are used with a scoring function to identify a subset of detections with spatial relationships that constitutes the most likely layout of detections. This subset of likely detections is used to adjust the scores associated with every 3D hypothesis box based on its 3D spatial relationships to objects in the subset.

In computing the weights, the spatial relationships of the true and false positive hypothesis boxes are used, not the relationships relative to the ground truth position of the objects. The ground truth positions are only used to identify true hypothesis boxes. Our goal is to differentiate between true and false detections, not ground truth and false detections.

7.2 Hypothesis Boxes

Each hypothesis box h_i has three components $h_i = \{b_i, y_i, s_i\}$ where b_i is an axis-aligned bounding box defined by two 3D points, $y_i \in 1 \dots K$ is a numeric label for K object types and $s_i \in \mathbb{R}$ is a detection confidence score. Hypothesis boxes are either produced by an object detector that works on 3D data or derived from the results of an object detector that operates in 2D. In our work, 3D hypothesis boxes are created from multiple 2D detections in images of the same scene using triangulation as we describe in Section 8.3.1.

We use axis-aligned bounding boxes which are efficient to compute from multiple 2D detections and allow for a wide range of spatial relationships. Indoor scenes generally have a dominant set of axes at right angles, defined by the surrounding walls, and objects are aligned with these axes. Therefore, our hypothesis boxes are also aligned with the dominant axes in the room. In scenes where the objects are not aligned, using axis-aligned boxes the hypothesis boxes can become exaggerated in the x and y axes with minimal effect on the

resulting qualitative spatial relationships. The box b_i could be replaced by any description of 3D space (e.g., triangle mesh, bounding sphere, etc.) which allows for the computation of 3D spatial relationships between hypothesis boxes.

Each hypothesis box has only one associated type and confidence score because that information corresponds to the output of most object detectors. An alternative model we considered first identifies candidate objects in the scene using approaches like table top segmentation, then applies N object detectors to those objects, producing a 3D region with N associated types and scores. The UBC team at the SRVC robot challenge [Meger et al., 2008] used this technique to increase object detection efficiency by allowing the robot to focus on specific areas and acquire high quality images. We did not pursue this approach as it requires the initial detection of candidate objects in the scene which was outside the scope of this work. The multiple object types and scores per hypothesis box representation used in the SRVC could be captured with our model, simply by having multiple overlapping hypothesis boxes for each type of object.

7.3 Scene Detection

Given the results in scene detection we demonstrated in Chapter 4, it should be possible, given multiple images of a scene, to determine its type with good accuracy. Thus, for this work, we assume that we know the type of scene and only detect objects likely to be found in that environment. Each scene type has its own model of spatial relationships found between the set of objects that are commonly found in that type of scene. A general model, that would encompass all commonly found objects, is beyond this work, but is discussed further in Chapter 9.

7.4 Model

Let $H_s = \{h_i : i = 1 \dots N\}$ be the input 3D hypothesis boxes produced for scene S. Let $Y = \{y_i : i = 0 \dots N\}$ be a label vector for all hypothesis boxes in a given scene. A background object label 0 is assigned to hypothesis boxes that are predicted to be false positives. A hypothesis box with a label other than background is called an *instanced* hypothesis box.

Hypothesis boxes are never relabeled with a new object type so y_i can only change between one object type and the background label. A detection should be relabeled only, it was

incorrectly labeled (i.e., the wrong detector activated on that point). Since detectors do not suppress each other, we rely on the correct object detector to also produce a detection at this location in space that overlaps in 3D with the wrong detection. Our approach then would adjust the scores of both detections to select the correct detection. This means there is no need to label detections to anything other than the type of their detector or as background to indicate that they are false positives. If only incorrect detectors fire on an object, we simply decrease the detection score on the incorrect detectors to suppress them and there is no true positive to increase.

We use the same scoring function as [Desai et al., 2009] which measures how a set of labels for hypothesis boxes agrees with our model of expected spatial relationships between objects. Maximizing the scoring function provides the most likely set of labels for all hypothesis boxes given their associated scores, types and spatial relationships. We define a scoring function for a set of labels Y on a set of hypothesis boxes H to be

$$S(H,Y) = \sum_{i,j} w_{y_i,y_j}^T d_{ij} + \sum_i w_{y_i}^T h_i \tag{7.1}$$

where w_{y_i,y_j} is a weight vector that encodes the value of two objects of type y_i and y_j sharing the spatial relationship d_{ij}. d_{ij} is a sparse binary vector that captures multiple spatial relationships. Since our relationships are not necessarily binary, a qualitative spatial relationship with N possible values would be mapped onto a binary vector of length N. These binary vectors are concatenated to create d_{ij}. The qualitative spatial relationships are based on the axis-aligned bounding volumes b_i and b_j. The seven relationships we use are described later in Section 7.7.

w_{y_i} represents a local detector score associated with hypothesis box h_i for object type y_i. It is made a two dimensional vector by appending a 1 to each detector score which allows this weight to be used to learn the bias between the object classes.

For the hypothesis boxes assigned to the background class, the local and pair-wise spatial relationship weights w_{y_i} and w_{y_i,y_j} are 0, effectively eliminating them from the scoring function. This means the model only considers the interaction between hypotheses that are considered to be true positives (instanced hypotheses). This helps both inference and learning by reducing the number of labels that influence the model.

7.5 Inference

In the inference step, the goal is to determine a set of hypothesis boxes that are true positives and then adjust the scores of all boxes based on their spatial relationships to this set. With correctly trained weights, the labels that constitute the most likely scene can be computed as $\arg\max_Y S(H, Y)$. Repeatedly computing $\arg\max_Y S(H, Y)$ is also an essential part of the learning algorithm, as we will explain in the next section, so efficiency of computation is essential.

7.5.1 Greedy Forward Search

Since computing $\arg\max_Y S(H, Y)$ is NP hard under most conditions, [Desai et al., 2009] use a simple greedy forward search. Let X be a set of 2D windows produced by an object detector on an image. Then let I be a set of instanced windows-class pairs (i, l) from X and let $Y(I)$ be the associated set of labels where $y_i = c$ for all instantiated window pairs in I and otherwise windows are the background class $y_i = 0$. Let $S(X, Y)$ be an equivalent scoring function to ours but applied to 2D windows rather than 3D hypothesis boxes. Let the change in score by instantiating windows i in I be

$$\Delta(i, c) = S(X, Y(I \cup \{(i, c)\})) - S(X, Y(I)) \tag{7.2}$$

Initialize $I = \{\}$, $S = 0$ and $\Delta(i, c) = w_c^T x_i$. The following greedy algorithm is then repeated:

1. $(i^*, c^*) = \arg\max_{(i,c)\notin I} \Delta(i, c)$
2. $I = I \cup \{(i^*, c^*)\}$
3. $S = S + \Delta(i^*, c^*)$
4. $\Delta(i, c) = \Delta(i, c) + w_{c^*,c}^T d_{i^*,i} + w_{c,c^*}^T d_{i,i^*}$

until $\Delta(i^*, c^*) < 0$ or all windows are instantiated. [Desai et al., 2009] report that this approach gave solutions close to the optimal on their 2D detections.

In our work on 3D scenes, we found greedy search failed to produce effective models during training. We determined that on our data it frequently does not compute the maximum scoring set of labels when compared with the the brute force solution. If the first selected detection is a false positive, it can throw off the entire rest of the search. This is more

noticeable in our work since, even if the first selected hypothesis box has a high associated detection score, it may not be well localized in 3D and will not have appropriate spatial relationships with other objects in the scene.

7.5.2 Branch and Bound Search

We replaced the greedy search approach with a branch-and-bound tree search which is slower but which provides an optimal solution given an upper bound on the number of possible true positive detections in the scene. Even if a better solution existing with more true positives, our approach is guaranteed to produce a solution at least as good as the greedy search because we initialize it with the greedy search solution. We treat the problem as a binary tree search where each node at depth d branches on instancing box I_d. Let Y be a set of labels for all hypothesis boxes. We maintain a set of viable candidate nodes C which can be represented as a label and node depth pair (d, Y). For computing the upper bound on a branch, we use:

$$Upper(H, Y, d) = \sum_{i>d} \sum_{j>d} U(i,j) + S(H, Y)$$

$$U_{i,j} = \begin{cases} 0 & \text{if } w^T_{Y_i,Y_j} d_{ij} + w^T_{Y_i} h_i \leq 0 \\ w^T_{Y_i,Y_j} d_{ij} + w^T_{Y_i} h_i & \text{if } w^T_{Y_i,Y_j} d_{ij} + w^T_{Y_i} h_i > 0 \end{cases} \tag{7.3}$$

Initialize $Y_{0...N} = 0$, $Y_{max} = Y$, $d = 0$, $c = (y, 0)$, $C\{c\}$ For each $c = (Y, d)$ in C:

1. $d = d + 1$
2. $Y_1 = (Y : Y_n = 0), Y_2 = (Y : Y_n = y_n))$
3. if $S(H, Y_1) > S(H, Y_{max})$ then $Y_{max} = Y_1$
4. if $S(H, Y_2) > S(H, Y_{max})$ then $Y_{max} = Y_2$
5. if $Upper(H, Y_1, d) > Y_{max}$ then $C = C \cup Y_1$
6. if $Upper(H, Y_2, d) > Y_{max}$ then $C = C \cup Y_2$

When computing $\arg\max_Y S(H, Y)$ in inference, efficiency can be gained by limiting the maximum number of instanced boxes in y. If a scene only contains a relatively small number of detectable objects, and once most of those have been identified, they supply sufficient spatial relationships to adjust the classification scores on all remaining true and false positive boxes.

7.6 Learning

In the training phase, our goal is to learn a set of weights such that Y after computing $\arg\max_Y S(H, Y)$ will instance only the true positive boxes.

7.6.1 Problem Formulation

Following the same formulation used in [Desai et al., 2009], wy_i, y_j the spatial relationship weight and w_y the local detector score's weight can be considered as a single weight w by rewriting the scoring function as:

$$S(H, Y) = \sum_{i,j} w_s^T \psi(y_i, y_j, d_{ij}) + \sum_i w_a^T \phi(h_i, y_i) \tag{7.4}$$

where w_s and $\psi()$ are vectors of length DK^2 and w_a and $\phi()$ are vectors of length $2K$, where D is the number of spatial relationships, where K is the number of object classes. The scoring function can then be rewritten as: $S(H, Y) = w^T \Psi(H, Y)$ where

$$w = \begin{bmatrix} w_s \\ w_a \end{bmatrix} \quad \Psi(H, Y) = \begin{bmatrix} \sum_i j\psi(y_i, y_j, d_{ij}) \\ \sum_i \phi(h_i, y_i) \end{bmatrix} \tag{7.5}$$

With this rewriting, inference is performed with the operation:

$$Y^* = \arg\max_Y w^T \Psi(H, Y) \tag{7.6}$$

For training the weights, we first need a collection of scenes where we have computed true and false positive hypothesis boxes. Given n scenes, we have hypothesis boxes H_n and labels Y_n. Next, we want to train a set of weights w such that a new set of hypotheses H_n tends to produce the correct set of labels $Y_N^* = Y_n$. We want a set of weights such that the true label H_n scores higher than all other possible labellings of the hypothesis boxes L_n.

The problem can be formalized as:

$$\begin{aligned} \arg\min_{w,\epsilon \geq o} \quad & w^T w + C \sum_n \epsilon_n \\ \text{s.t. } \forall n, H_n \quad & w^T \Delta\Psi(H_n, Y_n, L_n) \geq l(Y_n, L_n) - \epsilon_n \end{aligned} \tag{7.7}$$

where $\Delta\Psi(H_n, Y_n, L_n) = \Psi(H_n, Y_n) - \Psi(H_n, L_n)$. Since not all labellings are equally incorrect, we require a loss function which measures how incorrect a particular labeling L_n is

relative to the ground truth H_n and is used to penalize the slack value ϵ_n.

Determining the correctness of a box requires ground truth knowledge of the 3D extent of objects in the scene. The loss function we employ penalizes incorrect detections based on the degree of their overlap with a true positive. First we decompose the loss function as $loss(Y, L) = \sum_{i=1}^{N} l(y_i, l_i)$ over N hypothesis boxes. Then:

$$l(y_i, l_i) = \begin{cases} 1: & y_i \neq 0 \wedge l_i \neq y_i \\ 1: & l_i \neq 0 \wedge \nexists j \; s.t. \; [d(i,j) < 30 \; cm \wedge y_i = h_i] \\ 0: & \text{otherwise} \end{cases} \tag{7.8}$$

where $d(i, j)$ is the distance between the centroids of the hypothesis boxes i and j. The first case captures missing detections and the second case captures false positives with a check to ensure no true detection overlaps the hypothesis box. We use centroid distance rather than overlap, which is more commonly used to determine the success of 2D detectors [Desai et al., 2009], [Everingham et al., 2010]. Due to errors in camera position detection, we found that the hypothesis boxes we produced for small objects rarely overlap with their ground truth detections, even if the 2D detections that produced them were accurate. Similarly, large objects like refrigerators were much more likely to overlap with their ground truth.

7.6.2 Structured SVM Weight Training

Given the problem formulation in equation 7.7, w can be learned from training data using a structured support vector machine [Tsochantaridis et al., 2004] which we described in Section 2.2.3.

Learning a structured SVM requires are three functions: a loss function $\Delta(y, y')$, a feature mapping function $\Psi(x, y)$ and a function for computing the most violated constraints $\hat{y} = \arg\max_{y \in Y} H(y)$. For the loss function we use $loss(Y, L)$ from Equation 7.8. The feature mapping function $\Psi(x, y)$ computes a vector of all qualitative spatial relationships between non-background detections from y in scene x. In the computation of the most violated constraints, structured SVMs perform a computation very similar to Equation [7.1] [Desai et al., 2009]. We believe that a significant portion of the improvement provided by our branch and bound algorithm is the result of more accurately identifying these violated constraints. The degree of improvement can be seen in the experiment in Section 8.5.6.

7.7 3D Spatial Relationships

In our work, we use seven 3D spatial relationships for comparing object positions. Each qualitative relationship describes the position of a target hypothesis box T relative to a single reference hypothesis box R and the output is an integer value. We restrict ourselves to symmetric relationships so $F(R,T) = F(T,R)$, as this makes certain computations, like the branch and bound search, simpler to compute.

The size of the scenes we apply our model is *figural*, using the size terminology from Section 5.3.1, meaning they are similar in size to the human body and that can be perceived entirely without moving (e.g., a table or an office). Some of the scenes verge into the *vista* size, meaning they require a small amount of movement to examine fully but all are single rooms with no interior dividing walls. All the objects between which relationships are determined are represented by axis-aligned bounding boxes. Despite the fact that we operate in a physical space, hypothesis boxes do not obey physical constraints because they are the product of an object detector. Most noticeably, the hypothesis boxes can overlap each other arbitrarily and do not require physical support in the scene.

As we discussed in Chapter 5, there are four basic types of qualitative spatial relationships: distance, orientation, topology and shape. The relationships we use are based on a single type of spatial relationship. More complex features which combine multiple types of relationships we leave to future work. While our set of relationships is not exhaustive, we cover many of the qualitative spatial relationships available using axis-aligned bounding boxes.

7.7.1 Relationship Overview

Our two distance relationships are determined using $D(R,T)$, the shortest distance between the boundaries of the target and reference boxes, which is simple to compute for axis aligned bounding boxes. We use the distance between the boundaries rather than the centroids as it can more accurately capture spatial concepts like "touching".

One of the advantages of 3D detections is that accurate distances between detections can be determined. This is especially useful when the objects have predictable distances relative to each other as the partitions can capture human-centric concepts such as "within arms' reach". In 2D approaches, distances either require estimating the 3D geometry of the scene or rely on absolute measures that are based on metrics derived from the image (e.g., pixels or fraction of overall image).

The two distance relationships are:

- Absolute Distance: an interval-based comparison of the distance based on anthropometric values.
- Bounding Box Distance: a relative comparison of the distance based on size of the reference hypothesis box.

We have four orientation relationships that use a region-based interval algebra and have a frame of reference either using gravity to define the z axis or the shape of the room to define the x and y axes. The x and y axes are aligned with the dominant axes of the room geometry defined by the walls. While technically there are rooms with walls that do not form approximate right angles, we never encountered any in our data and they are relatively rare. In our descriptions below $min_{x,y,z}(h_i)$ and $max_{x,y,z}(h_i)$ are the minimum and maximum values in each axis of the axis-aligned bounding boxes for hypothesis i. $\Delta(b_R, b_T)_{x,y,z}$ is the overlap in the x,y or z planes of the target and reference box and $|b_i|_{x,y,z}$ is the length of hypothesis box h_i in the x,y or z axis.

The four orientation relationships are:

- Vertical Orientation: a comparison of the vertical partitions.
- Coplanarity: a comparison of the relative position of the tops and bottoms of the boxes.
- Vertical Alignment: a comparison that tests if the hypotheses form a vertical stack.
- Beside: a comparison of the overlap in the x and y axes that determines if the objects are aligned with the room walls.

The possible range of topological relationships is limited because we are using a bounding volume rather than the actual object shape. Also, since we are considering real, mostly convex, whole objects with no 3D part-based hypothesis boxes, objects would very rarely actually overlap each other. Therefore, we only use one simple topological relationship:

- Overlap: a comparison which tests if the two hypothesis boxes occupy the same space to a significant degree.

We have no shape or size relationships because we have no information about the shape of the objects in the hypothesis boxes. We do not have any useful size information either

because the approach we use for creating the axis-aligned boxes from 2D detections bases the size of the boxes on the type of the object, not the size of the 2D detections. If size is purely based on object type, then size comparisons are simply comparisons of type and not informative spatial relationships.

Below are formal descriptions of the seven spatial relationships we use and our reasoning behind their selection.

7.7.2 Absolute Distance

Our absolute distance relationship uses a consistent set of partitions to compare the distance between objects. The major challenge for absolute distances is determining the partitions on which to partition the distances. Terms such as *near* and *far* are relative to the objects and the environments, unlike terms such as *above* and *below*.

One reason absolute distances are appropriate for our work is that all the environments we are judging are on the same scale since absolute distances partitions need to be adjusted based on the size of the environment. We use the same partitions for both the kitchen and office scenes, 5 partitions with divisions at 10, 50, 100, 200 cm. These partitions were chosen using anthropometric measures to capture the concepts of *touching*, *within arm's reach*, *within one step* and *requires significant movement* [Tilley and Wilcox, 2002]. These relationships allow us to identify objects that are organized for different types of tasks. In an office, the structure of the room is centered around the desk and chair;f vital objects are placed within arms' reach (e.g., keyboard, mouse) while other less important ones (e.g., other chairs, printer) are further away.

It is possible to learn the partitions from the scenes but with limited training data it is not feasible to remove a significant number of scenes for learning the partitions. Using n-fold cross validation in our experiments left us with two alternatives: either we learn the partitions using all training and test scenes, which would mean training on test data, or learn different partitions based on the training set in each fold, which is impractical. To avoid these problems we opted instead for partitions we determined based on estimated anthropometric and environmental properties.

Let $P_{1...n}$ be increasing values that divide space into $n+1$ partitions. Our absolute distance

relationship outputs an integer value and is defined as:

$$AD(R,T) = \begin{cases} 1 & \text{if } D(R,T) < P_1 \\ i & \text{if } D(R,T) > P_{i-1} \wedge D(R,T) < P_i \\ n+1 & \text{if } D(R,T) > P_n \end{cases} \tag{7.9}$$

Bounding Box Distance

Bounding box distance is a relationship that uses a relative distance measure (a distance measure based on the size of the things being measured). The motivation for relative distance measures is that, when using descriptive terms such as *near* and *far*, the size of the objects in question is important. Two cups might be considered near if within 10 cm of each other, while two buildings near if they are within a 100 meters. In a kitchen or office, a similar effect may exist between large objects or small objects. Another advantage of relative distance is that there are no partitions to define or learn. One issue with relative distances is it is very coarsely granular, with only two possible states as we have defined the relationship. While it is possible to include additional partitions (e.g., half or double the size of the reference objects), these are not well motivated in the qualitative spatial literature.

In order to avoid odd effects resulting from objects that are much shorter in one axis than the others, we partition the distance between the objects using the longest internal span of either the target or reference objects. Using the maximum distance avoids the relationship being asymmetrical.

Let $Sp(h_i)$ be the longest internal span in hypothesis box i. Our box distance relationship outputs a binary value and is defined as:

$$BD(b_R, b_T) = D(b_R, b_T) < max(Sp(R), Sp(T)) \tag{7.10}$$

Vertical Orientation

Vertical orientation is an orientation relationship based on comparisons of the overlap of objects in the z axis and identifies semantic concepts like *above*, *below* or *level*. It relies on an external frame of reference that is defined by gravity and is therefore easy to determine. Vertical orientation is used in both 2D contextual work [Desai et al., 2009; Neumann, 2008; Ranganathan and Dellaert, 2007]. Vertical orientation relationships can be captured with

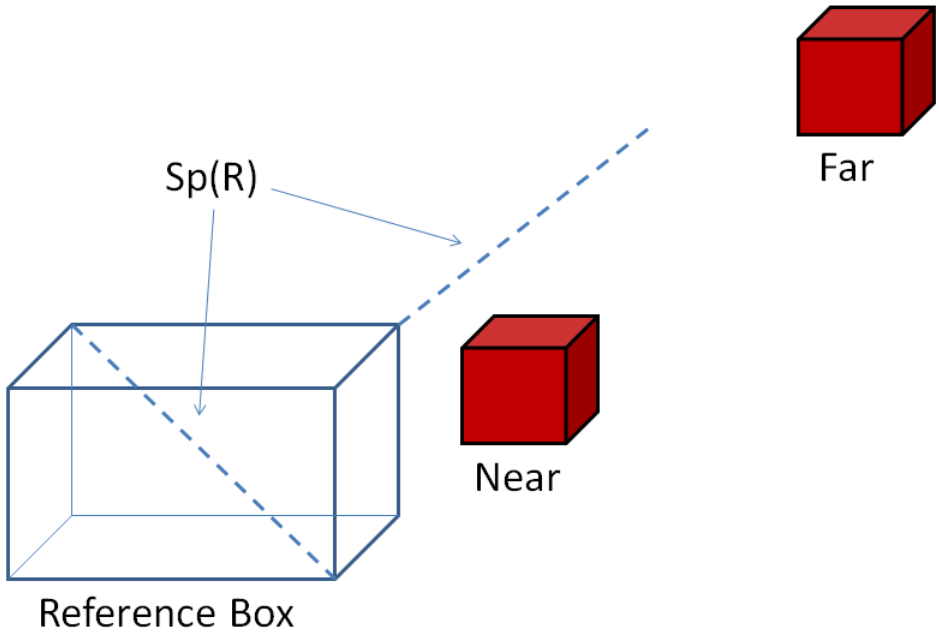


Figure 7.1: *This figure illustrates the box distance relationship. $Sp(R)$ and $Sp(T)$ (not shown) are the longest internal spanning vector of the reference box R, shown in blue), and target boxes T, shown in red. Since $Sp(R) > Sp(T)$ then $Sp(R)$ is used to define the maximum separation between objects that are "close".*

some accuracy in 2D as long as all images are taken from approximately the same height and angle but this becomes less effective as objects get closer to the photographer and the viewing angle becomes more downward tilted.

Vertical orientation comparisons of objects are effective because gravity is the most powerful factor controlling the layout of scenes. Every object we detect in our work is supported by a structure in the scene. Typically these are either the floor, a horizontal work surface like a desk or counter, or another object like a pot on a stove. Large objects are usually supported by the floor, while smaller objects are more readily available on work surfaces.

We use a modified version of the Allen interval algebra with a reduced number of states to make granularity of the relationship more coarse, as shown in Figure 7.2. Specifically, we removed those relationships that are defined by the top and bottom of the objects being

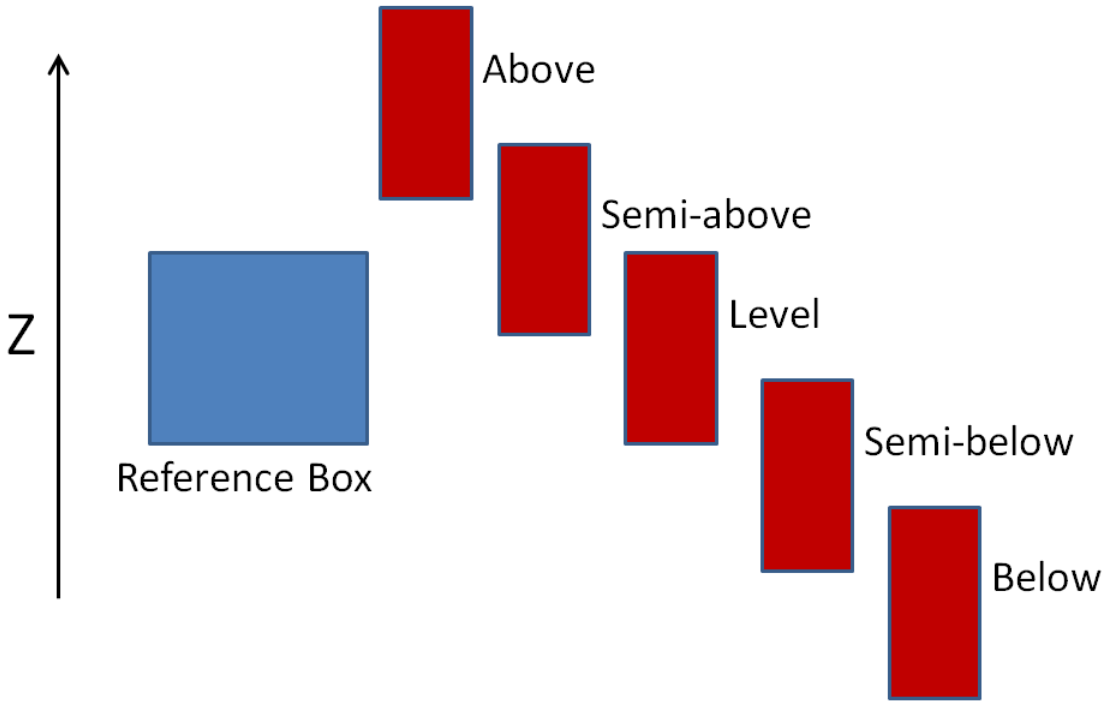


Figure 7.2: *This figure illustrates the vertical orientation relationship. The reference box R is shown in blue and different target boxes are shown in red with the resulting relationship beside them.*

close and moved them to a separate relationship called Coplanarity which we discuss next.

Our vertical orientation relationship has an integer output and is defined as:

$$VO(b_R, b_T) = \begin{cases} 1(\text{above}) & \text{if } max(R)_z < min(T)_z \\ 2(\text{semi-above}) & \text{if } max(R)_z < max(T)_z \wedge \\ & max(R)_z > min(T)_z \\ 3(\text{semi-below}) & \text{if } min(R)_z < min(T)_z \wedge \\ & min(R)_z > max(T)_z \\ 4(\text{below}) & \text{if } min(R)_z < max(T)_z \\ 5(\text{level}) & \text{else} \end{cases} \tag{7.11}$$

Coplanarity

Coplanarity is a relationship similar to the vertical orientation relationship but focuses on whether the top or bottom of objects are coplanar. It captures some of the relationships from the Allen interval algebra that were not included in Vertical Orientation. The relationship identifies objects that are either supporting or supported by other objects. It also identifies objects that are supported by the same scene structure like the floor or tables.

We wanted to remove the coplanarity relationships from Vertical Orientation as they are more affected by inaccurate localization and we want to be able to analyze their effectiveness separately later. Also Coplanarity cannot easily be determined from 2D data and has a distinct meaning semantically from concepts such as *above* and *below*.

Our coplanarity relationship has an integer output and is define as:

$$CO(b_R, b_T) = \begin{cases} 1(\text{same base}) & \text{if } |min(T)_z - min(R)_z| > \epsilon \\ 2(\text{same top}) & \text{if } |max(T)_z - max(R)_z| > \epsilon \\ 3(\text{supporting}) & \text{if } |max(T)_z - min(R)_z| > \epsilon \\ 4(\text{supported}) & \text{if } |min(T)_z - max(R)_z| > \epsilon \\ 5(\text{non-coplanar}) & \text{else} \end{cases} \tag{7.12}$$

These relationships are illustrated in Figure 7.3. *Same base* means that the underside of the two objects are approximately coplanar in the scene and therefore are likely on the same surface. *Same top* means that the top of the two objects are approximately coplanar, which occurs when both objects are integrated into a common work surface like a counter. *Supporting* and *supported* means that either the target or reference objects are aligned such that one might support the other.

The ϵ value accommodates inaccurate object localization in 3D (we used a value of 10 cm). Unlike in the other relationships, the coplanarity states are not mutually exclusive as objects can be both "same bottom" and "same top". We found this to be a rare relationship so we handled it by using integer value of the first true case but another state could be added without significantly changing the results.

It is important to note that Coplanarity is strictly based on the z axis partitions of the boxes. It does not consider the overlap in the x or y axes so it does not actually detect support relationships. Doing so would have required combining either topological and orientation or distance and orientation relationships. So, objects that are in a supporting relationship might be distant from each other.

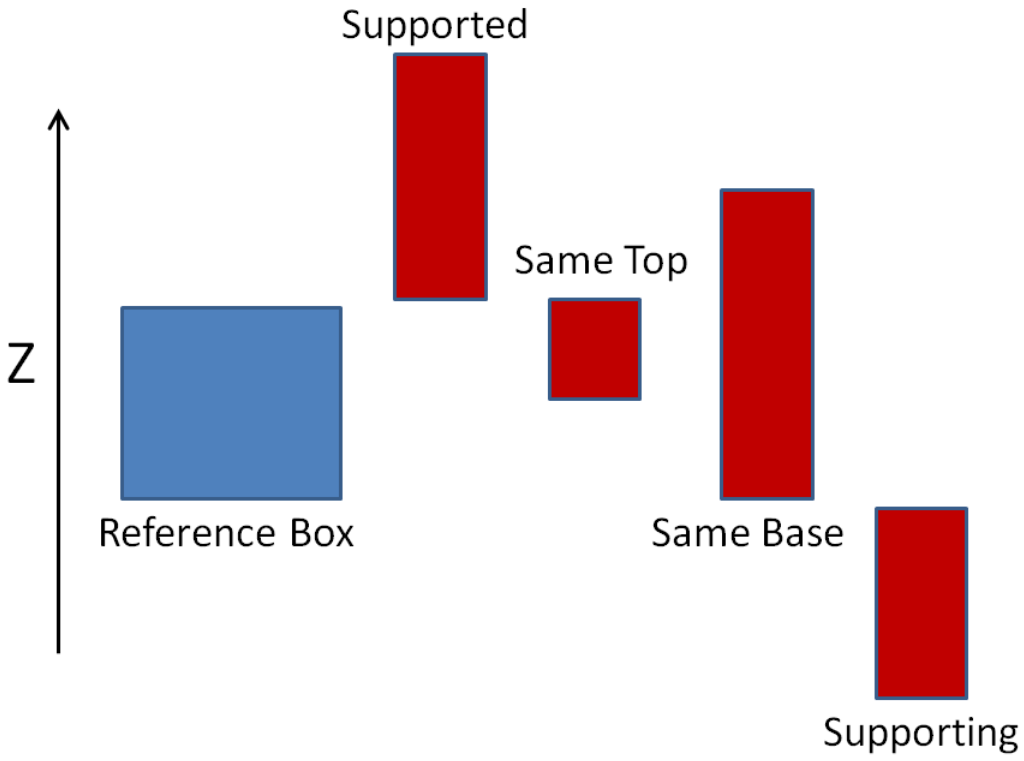


Figure 7.3: *This figure illustrates the Coplanarity relationship. The reference box R is shown in blue and different target boxes are shown in red with the resulting relationship beside them.*

Vertical Alignment

The vertical alignment relation is an orientation relationship that is used to identify objects directly over or under each other. Since actual relative elevations of objects are captured by the vertical orientation relationship, vertical alignment is based on the overlap in both the x and y planes as shown in Figure 7.4. This relationship is included to aid in determining the support relationships between objects which cannot be completely captured by the Coplanarity relationship.

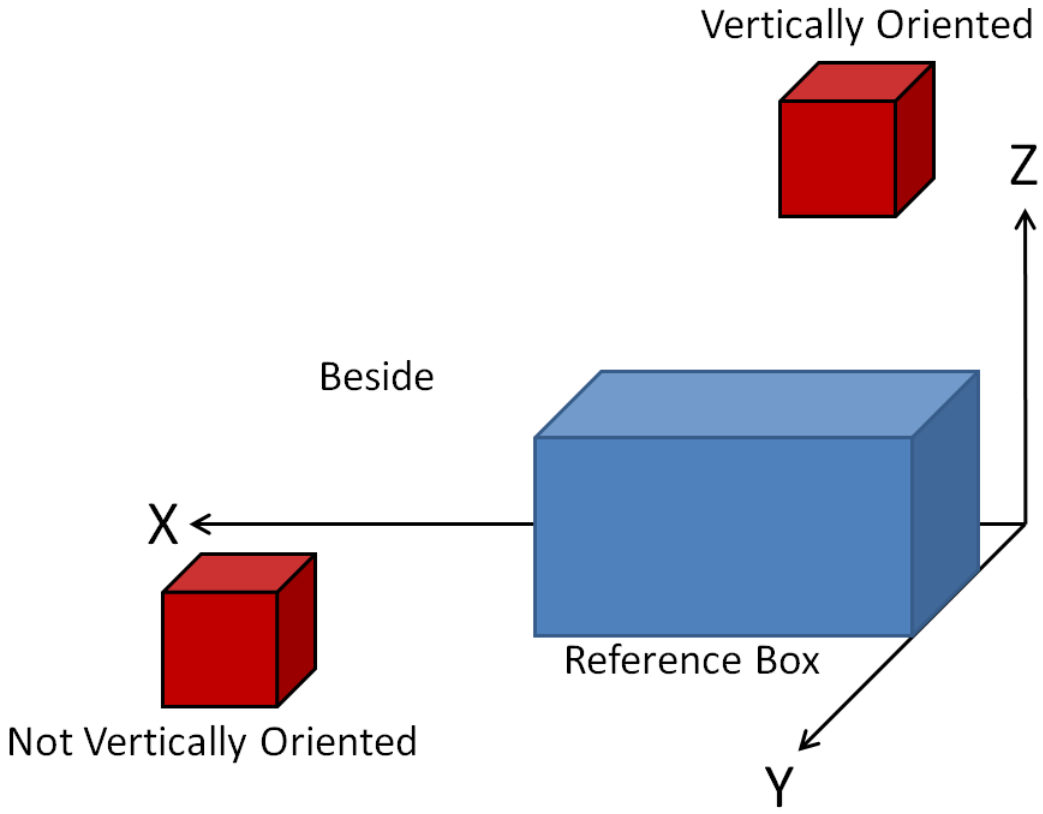


Figure 7.4: *This figure illustrates the vertical alignment relationship. The reference box R is shown in blue and different target boxes are shown in red with the resulting relationship beside them.*

Our vertical alignment relationship is:

$$VA(b_R, b_T) = \begin{cases} 1 & \text{if } \dfrac{\Delta(b_R, b_T)_x}{min(|R|_x, |T|_x)} > \lambda_{VA} \wedge \\ & \text{if } \dfrac{\Delta(b_R, b_T)_y}{min(|R|_y, |T|_y)} > \lambda_{VA} \\ 0 & \text{else} \end{cases} \tag{7.13}$$

where λ_{VA} is a variable defining the necessary potential percentage of overlap required (we used $\lambda_{VA} = 0.5$).

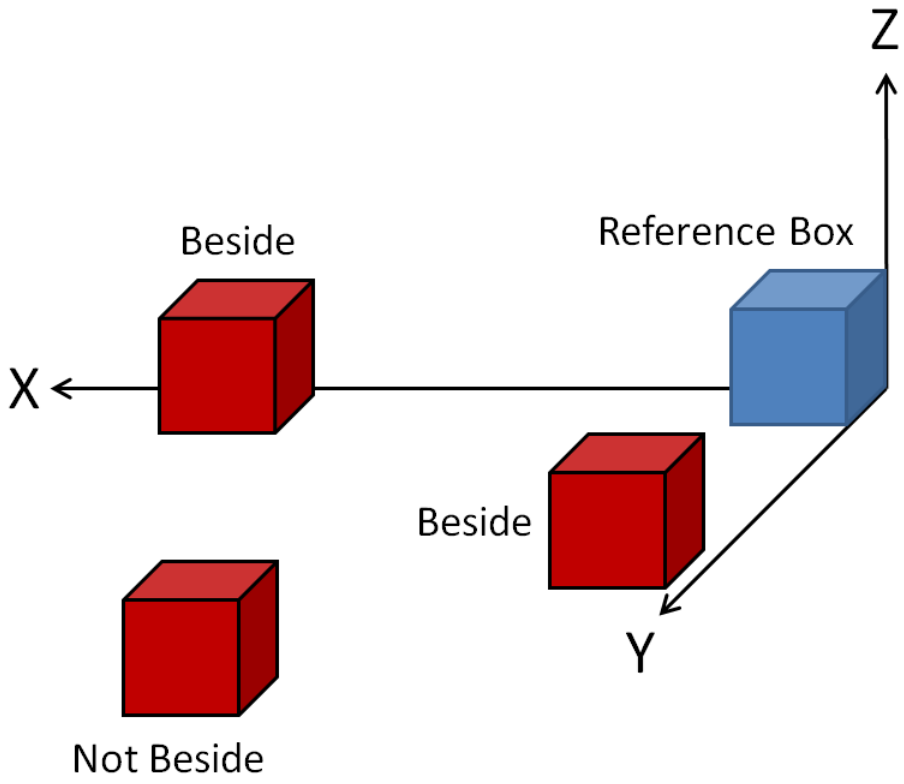


Figure 7.5: *This figure illustrates the vertical alignment relationship. The reference box R is shown in blue and different target boxes are shown in red with the resulting relationship beside them.*

Beside

The beside relationship is an orientation relation based on target and reference objects overlap in either the x or y axes, as shown in Figure 7.5. This novel relationship is based on our observation that, in indoor environments, objects are often aligned relative to a major axis in the environment. For example, in a kitchen a dishwasher, sink and microwave are either on or integrated into the same counter. This alignment can be for many practical reasons. Work surfaces and large objects are typically rectangular and therefore alignment with the walls minimizes wasted space. Also, keeping objects against the walls leaves the center of a room open for movement.

We use a constant $\lambda_B E$, which is the required percentage of overlap in each axes, to identify only boxes with a significant degree of overlap.

Our beside relationship has a binary output and is defined by:

$$BE(b_R, b_T) = \begin{cases} 1 & \text{if } \dfrac{\Delta(b_R, b_T)_x}{min(|R|_x, |T|_x)} > \lambda_{BE} \vee \\ & \text{if } \dfrac{\Delta(b_R, b_T)_y}{min(|R|_y, |T|_y)} > \lambda_{BE} \\ 0 & \text{else} \end{cases} \tag{7.14}$$

where λ_{BE} is a variable defining the necessary potential percentage of overlap required (we used $\lambda_{BE} = 0.5$).

Overlap

This simple topological relationship identifies when two detections occupy roughly the same space in 3D. It is primarily useful for handling mislabeled detections using an effect similar to non-maximal suppression [Neubeck and Van Gool, 2006], allowing an instanced detection to suppress another overlapping detection. It does not need the complex topological states found in the RCC-8 topological description, instead using a simple binary "overlap" or "disjoint" relation. We use the constant λ_BE, which is the required percentage of overlap in each axis, to identify only boxes with a significant degree of overlap.

Our overlap relationship has a binary output and is defined by:

$$OV(b_R, b_T) = \begin{cases} 1 & \text{if } \dfrac{\Delta(b_R, b_T)_x}{min(|R|_x, |T|_x)} > \lambda_{BE} \wedge \\ & \dfrac{\Delta(b_R, b_T)_x}{min(|R|_x, |T|_y)} > \lambda_{BE} \wedge \\ & \dfrac{\Delta(b_R, b_T)_y}{min(|R|_y, |T|_z)} > \lambda_{BE} \\ 0 & \text{else} \end{cases} \tag{7.15}$$

Chapter 8

Object Detection using 3D Spatial Relationships Results

In this chapter, we present experimental results for improving object detection scores using 3D spatial relationships and using the model described in Chapter 7. First, we present our method for acquiring training and test data on real world scenes. We describe our approach for computing 3D hypothesis boxes using 2D object detections from scene images and our model's effectiveness at improving the detection accuracy. We present experiments demonstrating our work on real-world data by applying our method to 3D hypothesis boxes created from 2D object detections.

After the real world experiments, we explore how our model would perform with different initial object detection results using simulation experiments where hypothesis boxes are generated from ground truth data (rather than from image data of the scenes). These experiments demonstrate how the properties of the hypothesis boxes (e.g., score, number of true/false detections, localization accuracy, etc.) affect the detection accuracy improvement from applying our model. Finally, we end with an analysis of the individual and cumulative effectiveness of the spatial relationships we described in Section 7.7 when combined with our model on simulated scenes.

8.1 3D Data Collection

Acquiring 3D data to train and test our model was a significant challenge. In Chapter 6 we used synthetic indoor scenes from a video game. While this was a good starting point for our work, it was difficult to prove that the object layouts were realistic and that our approach would work outside the scope of a game. Further progress required us to demonstrate our work on real world data.

Our video game data was models of entire houses but for our real world data we decided

to restrict our data to collecting image from individual rooms. This choice was based on the practical difficulties of collecting and integrating 3D views from different rooms into a single coherent model of object positions. Also, there is little evidence to suggest that humans make use of spatial relationships between objects in different rooms. This choice also allowed us to focus our model and choice of object detectors on a single type of scene at a time.

8.1.1 Fiducial Marker 3D Data Collection

The data for our experiments requires two elements. First is the ground truth 3D layout of objects in a scene, which is used to determine localization accuracy of the hypothesis boxes and in our simulation experiments to generate hypothesis boxes. Second is object appearances in the scene from which we produce 2D detections using an image-based object detector. These 2D detections are then used to triangulate a position in the scene of 3D hypothesis boxes for our real world experiments. In the real-world experiments we use image-based object detections but, alternatively, our experiments could be reproduced with range data and a shape-based object detector.

Both 3D ground truth positions and 3D hypothesis can be computed from images of a scene, as long as the camera position for each image is known, the images cover a variety of angles and there are multiple images of each object of interest.

Our data collection technique is based on the work of [Meger et al., 2011] for collecting 3D spatial data using a fiducial marker placed in the middle of each scene. A *fiducial marker* is a structure in an imaged space that is used as a point of reference for measuring distances, positions and scales. In our case, the fiducial used by [Meger et al., 2011] is a cube with a 3 by 3 grid ARTag markers [Fiala, 2005], bitonal planar patterns that encode a unique ID. The resulting grid of recognizable points allows us to determine camera position and scene scale relative to the marker.

Ground Truth 3D Object Data

This section describes how to compute $G_s = \{g_i : i = 1 \dots Z_s\}$, the ground truth boxes that describe the position and extent of each object in scene s. Each ground truth box consists of a pair $g = \{t, l\}$ where $t_i = (t^{min}_{xyz}, t^{max}_{xyz})$ is an axis-aligned bounding box defined by two 3D points and $l \in \{1 \dots K\}$ is a numeric label for the K types of object types in the scene we are detecting.

Figure 8.1: *A sample image of a kitchen taken from IKEA. The object in the foreground is a fiducial marker which allows us to recompute the camera position and integrate multiple images into a 3D model of the object layout.*

To determine g_i for each object, we photographed the scene from many angles, acquiring multiple images of each commonly found object and the fiducial marker (see Figure 8.1). Using the marker, we computed the camera position for each image. To compute the ground truth positions, using software developed by [Meger et al., 2011] we manually labeled 3D axis-aligned bounding boxes for commonly found objects in each scene. The labeling was performed by first manually identifying the center of each object in 3 or more images. We triangulated the 3D position of the centroid of the object from these points using singular value decomposition. Then an axis-aligned bounding box is projected onto each image and manually resized until it covers the object in each image. The resulting 3D boxes were checked using a 3D visualizer to determine that their layout corresponded to the scene.

The x and y axes of the scene are determined by the orientation of the fiducial marker which was placed to be aligned with the dominant axes of the room defined by the walls. While we did this manually during data collection, determining this basic scene structure can be done automatically and from a single image [Lee et al., 2010; Tsai et al., 2011]. We did not

use a robot for data collection as we were limited by what we could easily transport but our 3D data collection technique could be replicated by a robot equipped with a camera and performing simultaneous location and mapping (SLAM) [Montemerlo et al., 2003].

8.1.2 Locations & Scenes

There are obvious privacy issues involved in collecting data on the interior of houses. Additionally, houses typically only contain one example of most scene types. Finding enough volunteers and moving equipment to enough locations to acquire a broad dataset was infeasible if we used real houses. Therefore, we collected data in locations where we could capture many examples of the same type of scene. The first location was our department which provided many offices, kitchens and bathrooms. The second source of scenes was two IKEA stores that maintain many prefabricated scenes to demonstrate different selections of products. These scenes were ideal for data collection as they are realistic, show their contents well and are divided clearly into scene types (kitchens, offices, bedrooms, living rooms and bathrooms).

After data collection, we focused on kitchens and offices because we had the most examples and the scenes contained objects we determined were detectable by an image-based object detector with enough accuracy to provide a basis on which 3D qualitative spatial relationships could improve. Both bedrooms and living-rooms contained few objects that could be recognized by our detector. Objects like couches and coffee tables simply vary too much in appearance and lack strong defining visual characteristics. Finally, we had to discard bathrooms despite being able to detect some common bathroom objects reliably. In public bathrooms our data collection technique was ineffective because the bathrooms are designed to obscure views and hide areas from the users. In private bathrooms the area was often too small to effectively image with our approach since we need multiple shots from different angles of each object and the marker.

Our data set consists of 29 examples of kitchens and 27 examples of offices. For both scenes we selected a set of commonly found objects and labeled their positions in 3D. In the offices we selected keyboards, monitors, computer mice, chairs, telephones and printers. In the kitchens we selected sinks, faucets, dishwashers, ovens, microwaves, refrigerators, pans and small appliances. The small appliance category is an aggregation of electric kettles, toasters and coffee makers into a single category. With only 6-8 examples each of kettles, toasters and coffee makers across all kitchens, there were insufficient examples for learning. However, we observed they were roughly the same size and occupy similar locations in the scene so we aggregated them into the more general category of small appliance.

8.2 Experimental Overview

We evaluate our method experimentally on both simulated and real hypothesis boxes. We first present results on hypothesis boxes produced from image-based object detections from scene images. Then, our simulated hypothesis experiments use hypothesis boxes generated from ground truth data (rather than from image data of the scenes) and demonstrate how the properties of the hypothesis boxes affect the results of our method.

In all our experiments, we judge success by improvements in the average precision (AP) of all 3D hypotheses after we apply our method. AP is commonly used for comparing improvements in precision and recall [Everingham et al., 2010]. We compute AP by sampling the monotonically decreasing precision/recall curve at a fixed set of uniformly-spaced recall values $(0, 0.1, 0.2, \ldots, 1)$.

8.3 Image-based Hypothesis Experiments

The real world image-based experiments are designed to demonstrate the effectiveness of our model on 3D detections that are computed from actual 2D detections from images. The technique we use for determining 3D detections is just one of many possible approaches and does not take advantage of the 3D sensors that would likely be available on many robots. We are restricted to the data we could collect (i.e., images with a fiducial marker).

8.3.1 3D Hypothesis Construction

We use a technique for constructing 3D hypothesis boxes that combines multiple 2D image detection windows across multiple images of a scene. To produce our detections, we use the Deformable Parts Model detector [Felzenszwalb et al., 2008] trained with object images acquired from ImageNet [Deng et al., 2009].

The goal of this step is to construct a corresponding set of 3D hypothesis boxes $H = h_{1\ldots N}$. Given a set of 2D detections corresponding to the same object and using the camera positions we computed using the fiducial markers, we can triangulate the position of the object in 3D using the center point of each window. We use singular value decomposition to perform triangulation [Siciliano and Khatib, 2008]. To find the detections that correspond to the same object, we cluster detections using quality threshold clustering (QTC) [Jin and Han, 2010]. Because of occlusion, errors in detection and non-overlapping images, not every object in the scene will have a detection in each image. QTC selects a candidate 2D detection

Table 8.1: *The improvement in average precision (ΔAP) for objects and overall on 3D detections produced from scene images.*

Office	Ini. AP	Final AP	Δ AP	Kitchen	Ini. AP	Final AP	Δ AP
Keyboard	0.64	0.97	0.33	Sink	0.28	0.53	0.25
Monitor	0.67	0.60	-0.07	Faucet	0.68	0.64	-0.04
Lamp	0.45	0.62	0.17	Dishwasher	0.37	0.65	0.28
Mouse	0.80	0.61	-0.19	Oven	0.77	0.81	0.04
Chair	0.55	0.57	0.02	Microwave	0.65	0.74	0.09
Telephone	0.22	0.51	0.29	Refrigerator	0.35	0.24	-0.11
				Pot	0.43	0.47	0.05
				Small Appliance	0.64	0.56	-0.08
Overall	0.54	0.67	0.13	Overall	0.48	0.56	0.08

and iteratively adds the closest detection using a distance metric until the distance to the closest detection exceeds a threshold. Our distance metric is the error of triangulation in 3D.

To produce an axis-aligned box for each hypothesis, we use an average shape template based on the average dimensions of each object type in the training data. There are six possible poses for an axis-aligned box given a centroid. However, we assume all objects have an identifiable top and bottom (e.g., all microwaves are always found vertically aligned in the same direction) which reduces the possible poses to two. We reduce this to one pose by making the x and y axes of the template equal length. The template for an object type has a z length equal to the average height of ground truth boxes of that type. The width and length are made equal and set to the average length of the other two dimensions. Given typical error in triangulation, we found the effect on the resulting spatial relationships from making the length and width equal was minimal. Finally, we give each detection a confidence score equal to the average score of all 2D detections used to create it.

Table 8.1 presents the overall and per object improvement in average precision after applying our model for both types of scenes. On both types of scenes we saw improvement in the overall average precision almost double that observed in [Desai et al., 2009] though of course the data sets are different. Also, like them, we observed variation in the degree of improvement from class to class with results on some classes becoming worse. As we will discuss in our next section on synthetic detection experiments, we have observed that this tendency to improve on some classes and not others is caused by lower numbers of true positive detections in the training and test data and inaccurate localization in the detectors. In our work, image-based detections produce hypothesis boxes for approximately $60 - 70\%$ of all detections and many were poorly localized. Since the goal of this work is

to demonstrate how effectively 3D spatial relationships can improve object detections, we move onto our synthetic detection experiments.

8.3.2 Instanced Hypothesis Boxes of Image-based Experiments

This section provides examples of which 3D hypothesis boxes were instanced in some of our real world experiments. Instanced hypothesis boxes are the high confidence subset of boxes that are used to adjust the confidence scores of all boxes to remove false positives. Instanced boxes comprises both high confidence boxes and those that constitute a layout consistent with our model (as selected by the branch and bound tree search). A good set of instanced hypothesis boxes should be well localized.

Figures 8.2 through 8.13 show both good and bad examples of instanced hypothesis boxes and the demonstrate both the benefits and potential problems of using 3D spatial relationships for improving object detection. These are discussed in greater detail in the captions of each figure.

It is important to remember that these configurations of boxes are used to adjust the final scores of all detections, they are not the end result of our approach. Even if they are based on false positive detections, as long as they constitute a likely configuration, they can still be used for adjusting the scores of other detections. If a hypothesis boxes is the result of a misclassification but is still well localized, it can still be useful for adjusting the confidence scores of the other boxes. For example, if a stove was misclassified as a dishwasher but still well localized, that box could still provide valuable spatial information about the structure of the room and counter. Similarly, a poorly localized hypothesis box might still be useful because some of our spatial relationships, like distance, vertical orientation and beside are coarse enough to be tolerant to poorly localized detections.

8.4 Generated Hypothesis Experiments

Our image-based hypothesis experiments demonstrate a fully functioning system for performing object detection and localization in 3D with spatial relationships used to improve detection accuracy. However, there are many approaches to the creation of 3D hypotheses: other 2D detectors that could be used for triangulation, different techniques for producing 3D boxes for 2D and a whole range of object detection techniques that perform object detection on 3D data. How might other 3D detectors utilize 3D spatial relationships to improve their detection rates?

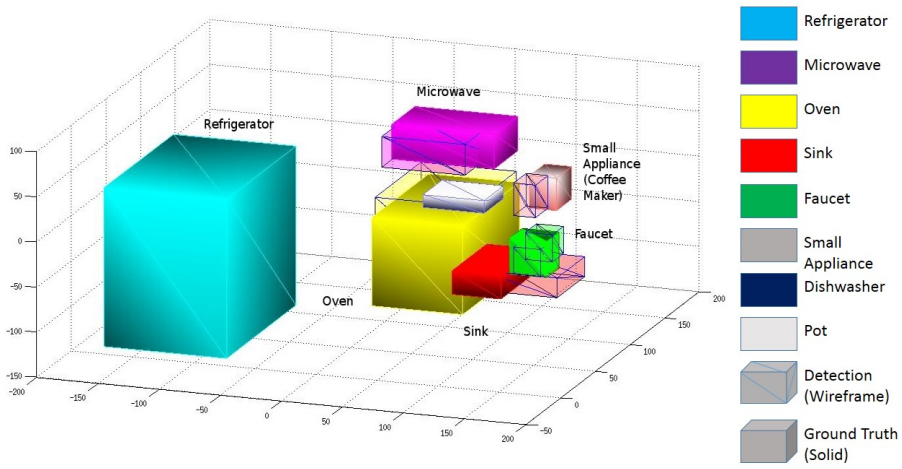


Figure 8.2: *An illustration of good instanced hypothesis boxes from image-based detections for a kitchen scene. In this scene most of the objects had well localized boxes that comprised a likely layout so most objects overlap an instanced hypothesis box.*

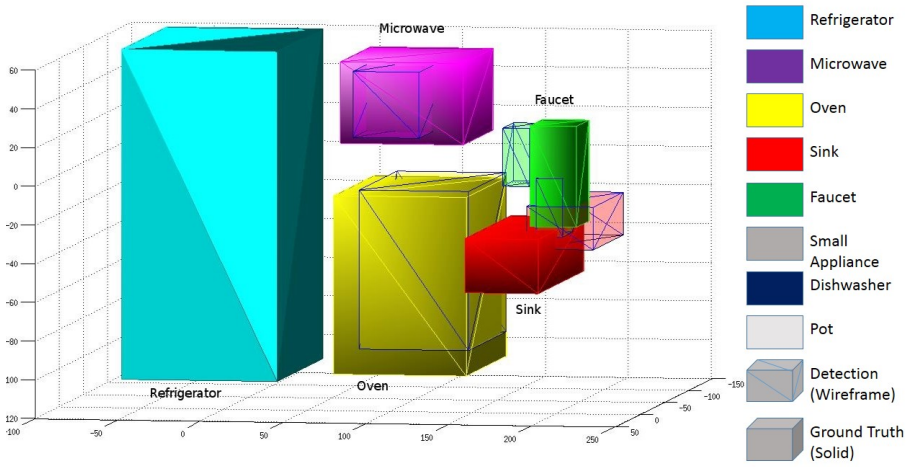


Figure 8.3: *An illustration of good instanced hypothesis boxes from image-based detections for a kitchen scene. Again, many of the objects in the scene overlap an instanced hypothesis box. There are multiple overlapping boxes included for the faucet because both boxes are well localized enough to be considered correct boxes and our loss function does not penalize additional good boxes.*

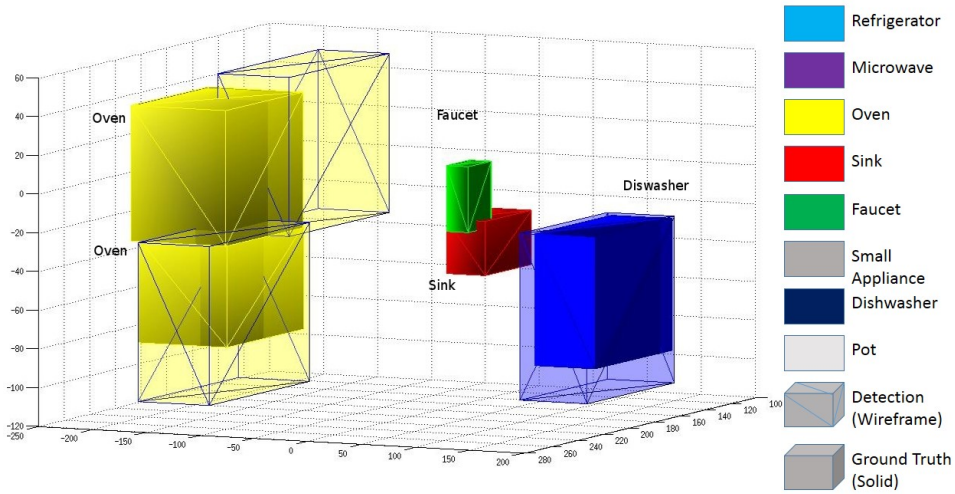


Figure 8.4: *An illustration of good instanced hypothesis boxes from image-based detections for a kitchen scene. This scene has fewer instanced hypothesis boxes but contained an example of stacked ovens being correctly detected and instanced. The localization of the top oven box is poor but it is still instanced because it shares spatial relationships with the bottom oven boxes observed in other stacked ovens in the data set.*

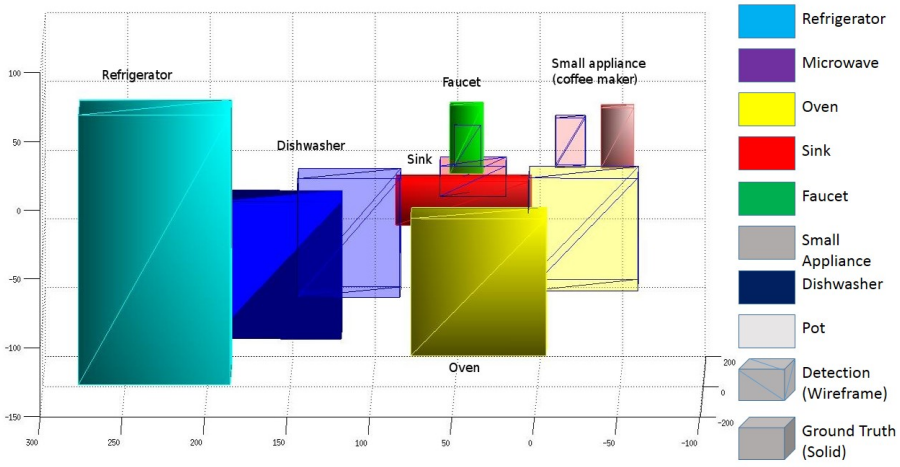


Figure 8.5: *An illustration of bad instanced hypothesis boxes from image-based detections for a kitchen scene. The model can select multiple poorly localized hypothesis boxes because they constitute a likely layout. Here, the oven, dishwasher and coffee maker boxes were likely selected because the oven and dishwasher top and coffee maker bottom are coplanar, a layout common in many scenes.*

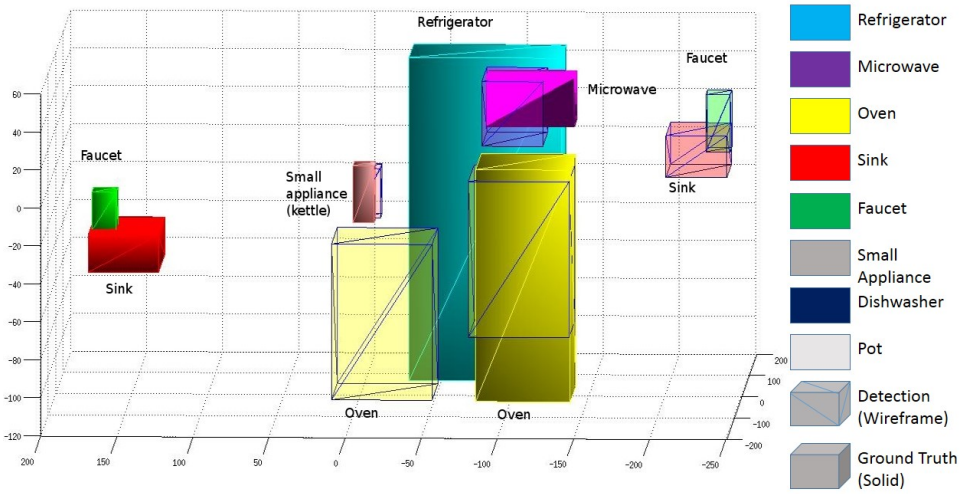


Figure 8.6: *An illustration of bad instanced hypothesis boxes from image-based detections for a kitchen scene. This scene contains two instanced oven hypothesis boxes, side by side, with one poorly localized and one well localized. The extra oven box might have been included because multiple oven boxes near each other are common in stacked ovens and the extra detection is close to a small appliance detection.*

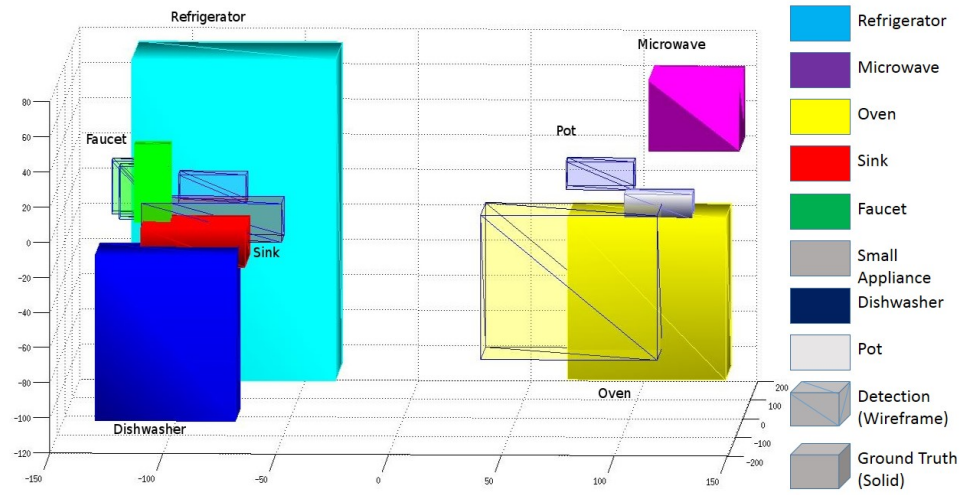


Figure 8.7: *An illustration of bad instanced hypothesis boxes from image-based detections for a kitchen scene. In this scene, both the oven detection box and pot detection box are significantly offset from their ground truths. Pots are often observed in the data set on top of ovens so the model selected the pot and oven boxes that shared that relationship.*

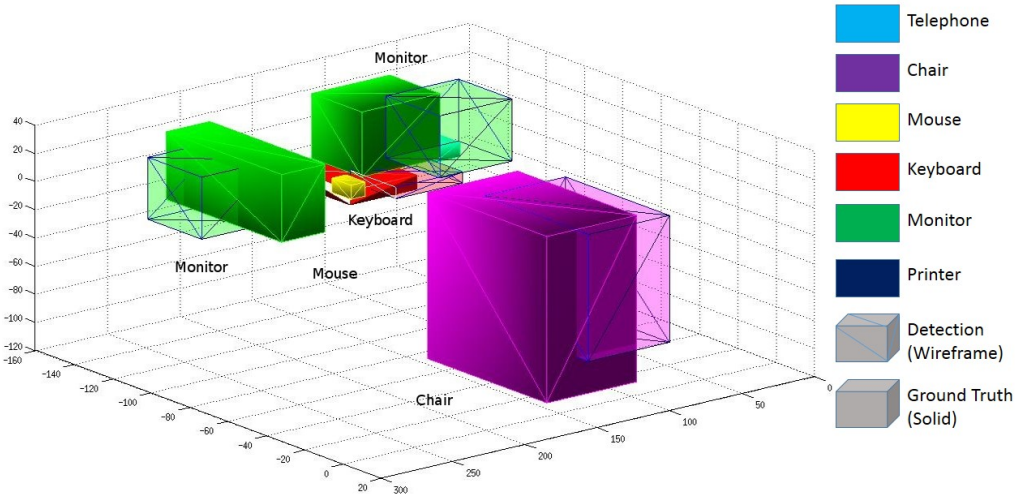


Figure 8.8: *An illustration of good instanced hypothesis boxes from image-based detections for an office scene. In this scene several objects had well localized boxes that comprised a likely layout so many objects overlap an instanced hypothesis box.*

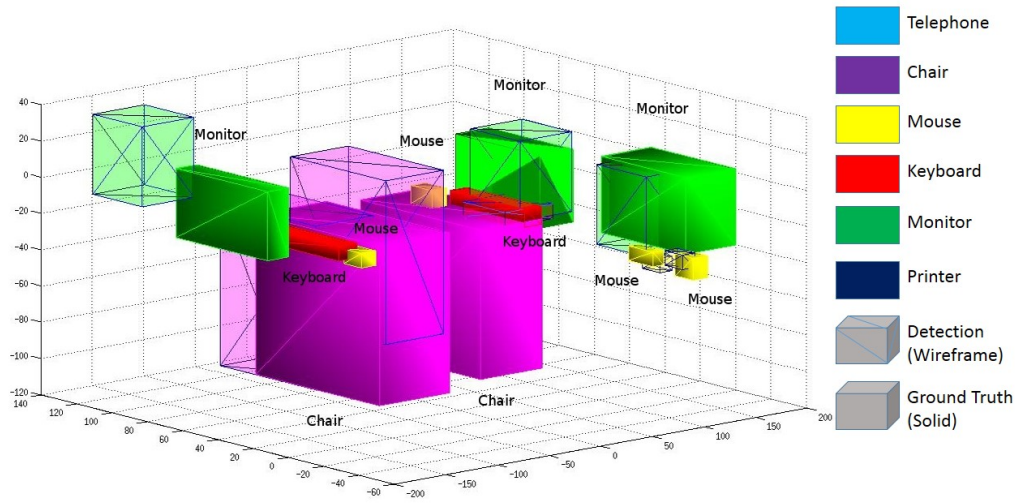


Figure 8.9: *An illustration of good instanced hypothesis boxes from image-based detections for an office scene. This scene is very complicated with a large number of objects detected and localized accurately. Not every object is well localized, one of the monitors is offset significantly. Also, the two close chairs can lead to extra poorly localized boxes when 2D detections from different chairs are combined together to create a 3D hypothesis box.*

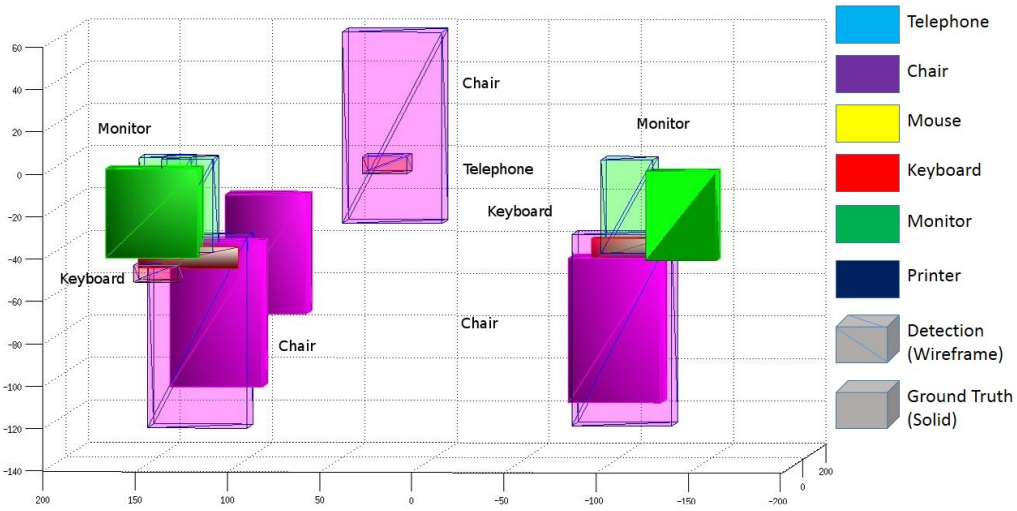


Figure 8.10: *An illustration of good and bad instanced hypothesis boxes from image-based detections for an office scene. Two parts of the scene contain well localized detections on either side and an extra chair and telephone false positive detections appear in the background.*

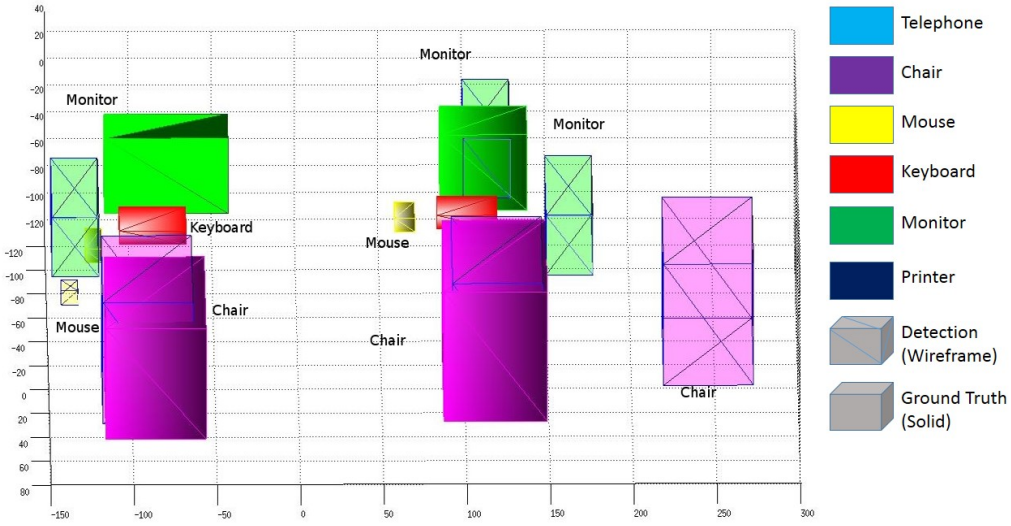


Figure 8.11: *An illustration of good and bad instanced hypothesis boxes from image-based detections for an office scene. Both chairs and one monitor hypotheses were well localized but the two false positive monitors and one chair were added, though they do comprise a reasonable scene layout. The monitor on the left was likely added because there is a mouse hypothesis next to it.*

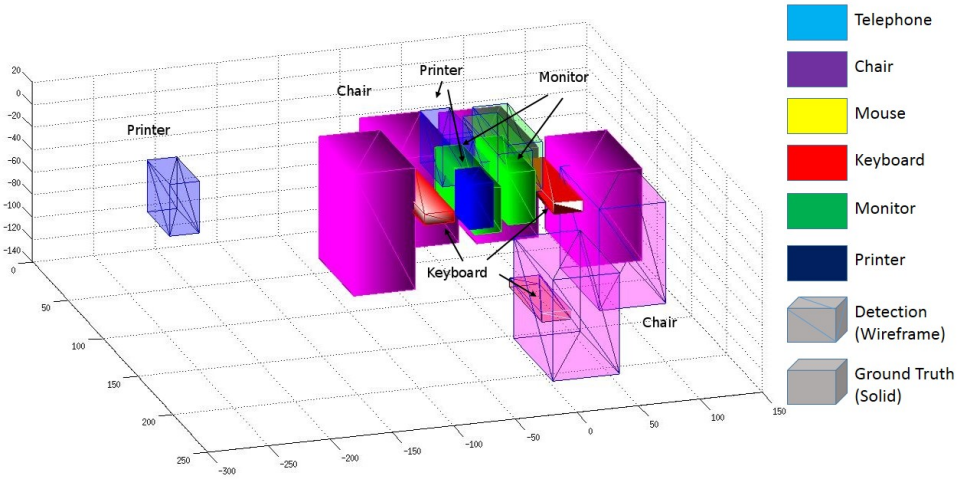


Figure 8.12: *An illustration of bad instanced hypothesis boxes from image-based detections for an office scene. The problem in this scene was too many false positive and badly localized boxes. The scene had a horseshoe arrangement of desks with the fiducial marker placed in the middle and many objects close together. This lead to many incorrectly localized and false positive boxes because every camera frustum overlapped, meaning many more intersections in the 3D hypothesis box creation.*

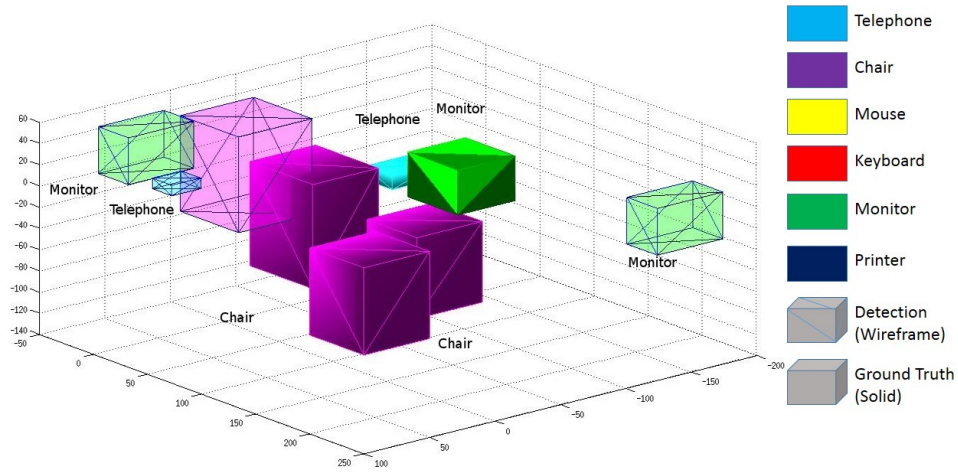


Figure 8.13: *An illustration of bad instanced hypothesis boxes from image-based detections for an office scene. In this scene there simply were not enough images taken and there are few 2D detections. This resulted in few hypothesis boxes and so only a small number of poorly localized boxes were instanced.*

To test this question and determine the effectiveness of our model for improving detection rates, we perform a broad range of experiment which simulate the creation of 3D boxes. In the broader scope of this thesis, these experiments allow us to examine how the properties of the hypothesis boxes affect the overall effectiveness of our approach. They provide an upper bound for how effective 3D spatial relationships could be at improving detection accuracy. Furthermore, they identify how the properties of the hypothesis boxes affect detection rates, allowing future work to optimize their inputs for our model.

8.5 Simulated Hypothesis Experiments

In our simulated hypothesis experiments we simulate the results of an object detector and produce the 3D hypothesis boxes algorithmically for both training and test data. *Simulated hypotheses* are created from ground truth detections by shifting the position to simulate localization error and providing a detection score from a distribution. Different techniques are used to create true and false positive detection positions and scores. Simulated experiments examine how hypothesis boxes properties affect our approaches effectiveness. The properties examined are:

- Classification Score
- Number of true detections
- Localization error
- Branch and Bound vs. Greedy Search

We also present experiment results comparing the effectiveness of individual 3D spatial relationships we described in Section 7.7 on different types of scenes and varying levels of localization accuracy.

8.5.1 Simulated True Positive Detections

Simulated true positive detections correspond to a detector identifying an object in the scene correctly. The resulting score and localization will vary but we constrain the localization error such that the detection is correct according to the loss function 7.8. Given a ground truth detection box $g_i = \{t_i, l_i\}$, a true positive hypothesis box $h_i = \{b_i, y_i, s_i\}$ is defined

by:

$$
\begin{aligned}
&b_i = \{t_i^{min} + \{D_x, D_y, D_z\}, t_i^{max} + \{D_x, D_y, D_z\}\} \\
&\ s.t.\ \sqrt{D_x^2, D_y^2, D_z^2} < 30\text{cm where } D = \mathcal{N}(0, \sigma_{box}) \\
&y_i = l_i \\
&s_i = \mathcal{N}(-0.5, 0.25) + T
\end{aligned}
\tag{8.1}
$$

$\mathcal{N}(0, \sigma_{box})$ is a normal distribution and σ_{box} is used to control the degree of localization accuracy. Without actual image detections to provide a detection score for the hypothesis boxes, we generate scores from a distribution. $\mathcal{N}(-0.5, 0.25)$ is a normal distribution and T is a constant we add to positive detections to control the average difference between true and false positive detections. The values $(-0.5, 0.25)$ were chosen to resemble the distribution of scores with the 3D object detector we describe Section 8.3.1. Changing T allows us to simulate detectors with varying ability to differentiate true positives and false positives.

8.5.2 Simulated False Positive Detections

Simulated false positive detections come in two types based on our observations of how false detections are produced in 2D image detectors and our results when creating 3D hypothesis boxes from image detections. False positives of the first type are similar to false positive detections generated by background objects or hypothesis boxes with extremely high localization error. These detections generally appear to be randomly positioned in the scene. False positives of the second type are the results of detector mislabeling. These detections correspond to a detector firing on an object of the wrong type which happens for objects with similar appearances such as microwaves and ovens. Both types of false positives are created from a reference ground truth detection $g_i = t_i, l_i$.

$$
\begin{aligned}
&\text{Background False Positive} \\
&b_i = \{t_i^{min} + \{U_x, U_y, U_z\}, t_i^{max} + \{U_x, U_y, U_z\}\} \\
&\ \text{where } U = \mathcal{U}(0, \sigma_{scene}) \\
&y_i = l_i \\
&s_i = \mathcal{N}(0.5, 0.5)
\end{aligned}
\tag{8.2}
$$

$$
\begin{aligned}
&\text{Mislabeled False Positive} \\
&b_i = \{t_{xyz}^{min} + \{D_x, D_y, D_z\}, t_{xyz}^{max} + \{D_x, D_y, D_z\}\} \\
&\ s.t.\ \sqrt{D_x^2, D_y^2, D_z^2} < 30\text{cm where } D = \mathcal{N}(0, \sigma_{box}) \\
&y_i = l_{rand} s.t. l_{rand} \neq l_i s_i = \mathcal{N}(0.5, 0.5)
\end{aligned}
$$

$\mathcal{U}(0, \sigma_{scene})$ is a uniform distribution where σ_{scene} is a approximate value for the average size of all scenes. l_{rand} is a random integer from $(1 \ldots K)$.

8.5.3 Simulated Hypothesis Experimental Procedures

We control the number of true detections per scene with P_{det}, the probability that each ground truth object has an associated true detection. We generate F_{back} background false positive detections and F_{mis} mislabeled false positive detections for each ground truth object.

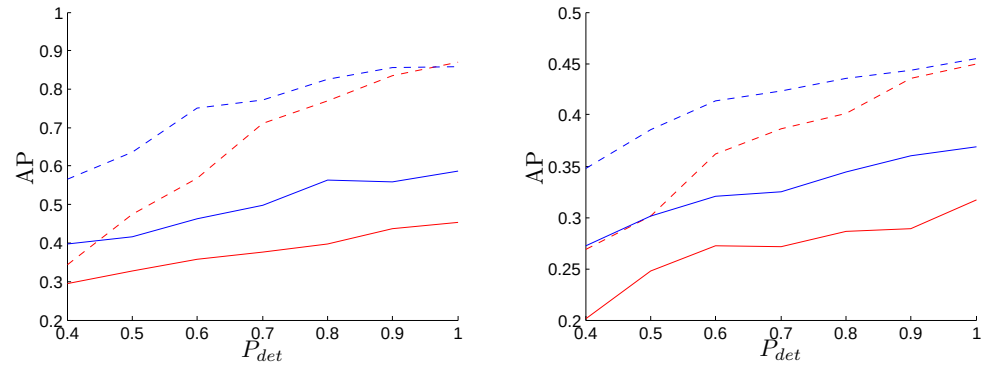
For each experiment we produce ten different layouts of the hypothesis boxes for each scene. We train and test our method using 5-fold cross validation across the scene. Results shown are the average precision of the true hypothesis boxes before and after using spatial relationships to modify detection scores. We average the AP over the ten layouts. In each experiment we vary a set of parameters. Unless otherwise indicated, each experiment uses the following constants: $\sigma_{box} = 20$cm, $T = 0.2$, $\sigma_{scene} = \{300, 300, 200\}$, $P_{det} = 1$, $F_{back} = 2$ and $F_{mis} = 1$.

8.5.4 Detection and Localization Error Experiment

This experiment is designed to determine the combined effects of detection and localization error. Detection errors occur when the object detection does not produce a hypothesis box for an object in the scene. Having enough objects detected in both training and test data is important because each additional good hypothesis box provides information on spatial relationships to all other boxes. This means the number of informative spatial relationships grows exponentially in the number of detection boxes. Localization error occurs when an object is detected in the scene but the hypothesis box does not completely overlap the ground truth. Some spatial relationships are sensitive to poor localization, such as vertical alignment and coplanarity especially, but can be highly informative.

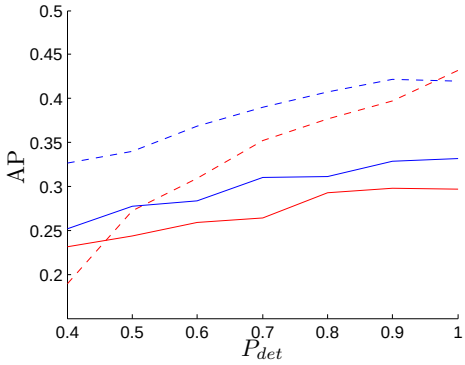
Figure 8.14 shows how the average precision improvement increases as a function of decreasing σ_{box} and increasing the number of true detections (controlled by P_{det}). As σ_{box} increases, the negative effects of having fewer detections grows. Without localization error, we see a very large improvement from our method but perfect localization is unlikely in real world results. However, our method works well with either poorly localized detections or missing detections. The combination can cause our method to under perform.

Clearly offices perform better with decreasing object counts and this could be because spatial

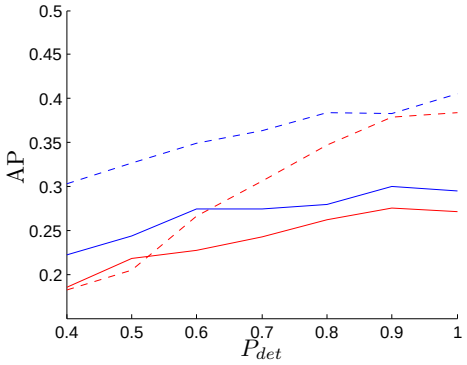


(a) $\sigma_{box} = 0$

(b) $\sigma_{box} = 10$

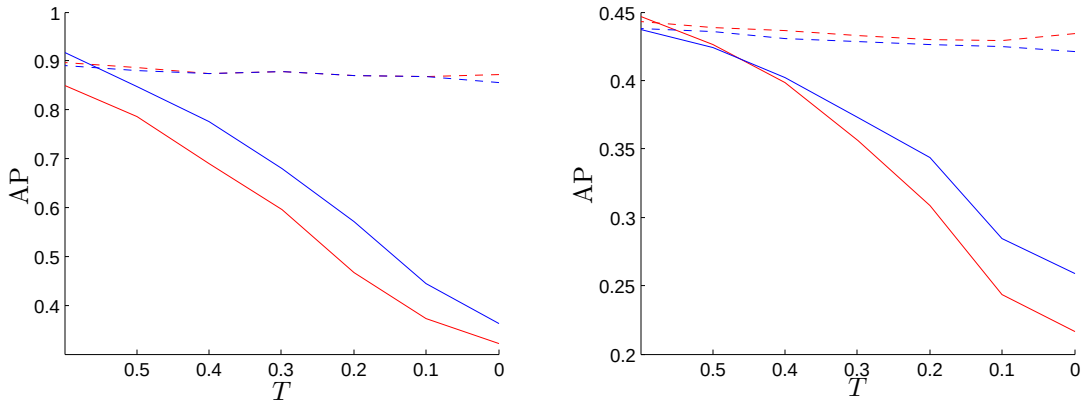


(c) $\sigma_{box} = 20$



(d) $\sigma_{box} = 30$

Figure 8.14: *Simulation results which vary the number of true positive detections, controlled by P_{det}, the probability that each ground truth object has an associated positive detection. Results are shown at different levels of localization error (controlled by σ_{box}). Kitchen results are in red and offices in blue. Solid lines show average precision before applying our method and dotted lines after. Our model provides a significant improvement with either good localization or many positive detections.*



(a) $\sigma_{box} = 0$ **(b)** $\sigma_{box} = 20$

Figure 8.15: *Simulation results which vary T, the average difference between the true and false positive scores, shown at different levels of localization error (controlled by σ_{box}). Kitchen results are shown in red and offices in blue. Solid lines show the average precision before applying our method and dotted lines after. The largest improvement in average precision occurs when T is small, simulating a detector with a poor ability to differentiate true and false positive detections.*

relationships between office objects are more consistent. However, we believe that offices perform better as the rate of detection decreases because there are on average twice as many objects in office scenes, even though there are fewer classes of objects being detected. Many of the offices have multiple chairs, monitors, keyboards, etc., while most of the objects found in the kitchens only occur once in a scene. Overall, having enough true positive detections is more important than having them accurately localized as our 3D spatial relationships can handle poor localization but without true positives there are not enough relationships for learning.

8.5.5 Score Separation Experiment

This experiment is designed to test how the difference in average scores between true and false positive detections affects the improvement in average precision. The difference in scores is controlled by the T variable. Large values of T simulate an object detector that is effective at differentiating between true and false detections. Since we found localization accuracy had such a significant effect in the previous experiments, we ran this experiment varying both T and σ_{box}.

Figure 8.15 shows the improvement in average precision from our method increases as the difference in score between true detections and false positive detections decreases. The improvement in AP increases because our method works best when there is confusion between true and false positives. As the difference in scores decreases, our method's performance remains consistent but the average precision before our method is applied decreases.

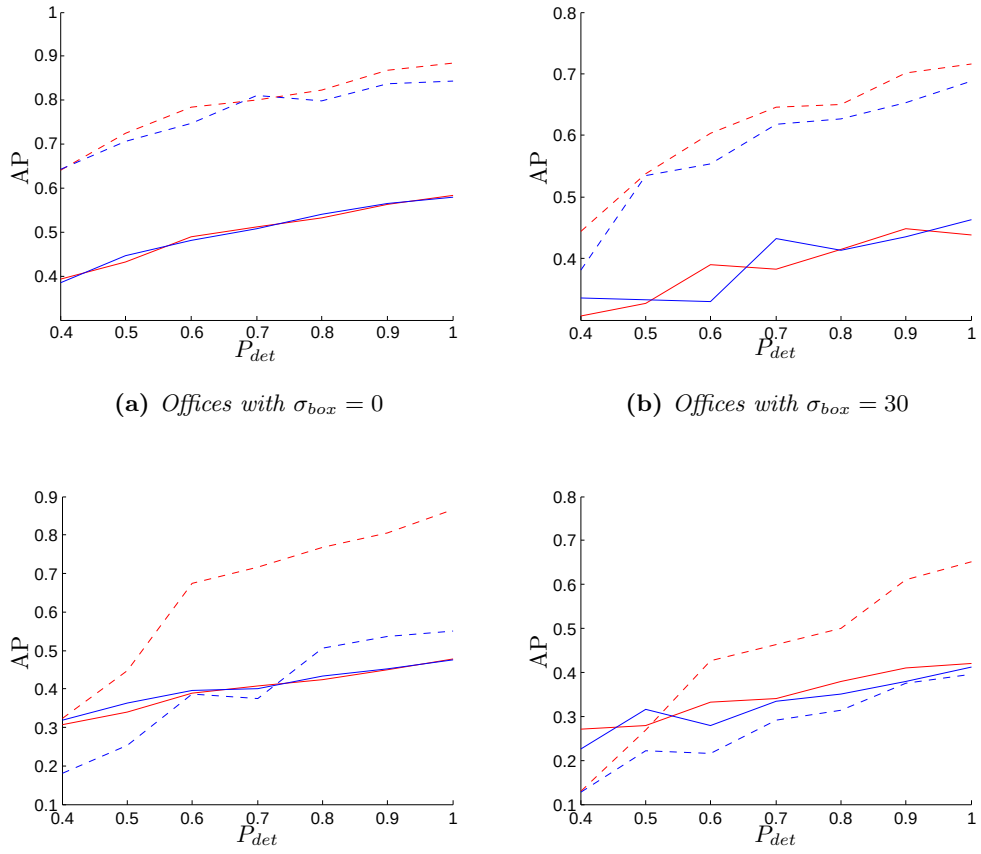
This ability to function well regardless of score difference has several advantages. Firstly, we are tolerant to object classifiers that either do not produce a margin of classification or ones where that margin is not very informative. Secondly, we did not need a more elaborate technique for combining together scores from 2D detections into 3D. As we described in Section 8.3.1, we use the average of the 2D detections for the hypothesis box score. We considered using a more complex technique, such as learning an SVM that would recompute the score based on the 2D detection scores, the number of detections used, the error in triangulation and other scene properties. However, it is clear that it is unnecessary to provide accurate initial confidence scores for our approach to be effective.

8.5.6 Branch and Bound Tree Vs Greedy Search

As we discussed in Sections 7.5.2 and 7.6.2, we implemented a branch and bound tree search approach for both inference and structured SVM training. This approach replaced the greedy search based approach used in [Desai et al., 2009]. This experiment is designed to examine the improvement in average precision we derived from using a branch and bound tree search instead of the simpler greedy search approach.

Figure 8.16 shows how our branch and bound search outperforms the greedy search approach. The effect is greater when there is more error in localization. This is likely due to branch and bound being less likely to be misled by a poorly localized, highly scored detection, which the greedy search approach can fixate on early while the branch and bound finds an overall more optimal solution.

The most evident result is that branch and bound also performed more consistently across scene types, with the greedy approach performing negatively on the kitchen data when localization or number of positive detections are low. These negative results for greedy search are either because the spatial relationships in the kitchen are harder to identify, (possibly because they are less structured than the offices) or because there are fewer objects, and therefore relationships between positive detections, for learning. Either way, branch and bound search seems better able to make use of the available training data and produces a better model.



(a) *Offices with* $\sigma_{box} = 0$

(b) *Offices with* $\sigma_{box} = 30$

(c) *Kitchens with* $\sigma_{box} = 0$

(d) *Kitchens with* $\sigma_{box} = 30$

Figure 8.16: *Simulation results that compare the effectiveness of our branch and bound search against the [Desai et al., 2009] greedy search. We varied the localization error controlled by* σ_{box}*. Branch and bound results are in red and greedy search results in blue. Solid lines show average precision before applying our method and dotted lines after. From these it is clear that the branch and bound approach improvements vary between scene types but did provide consistently better results.*

8.5.7 Spatial Relationship Comparison

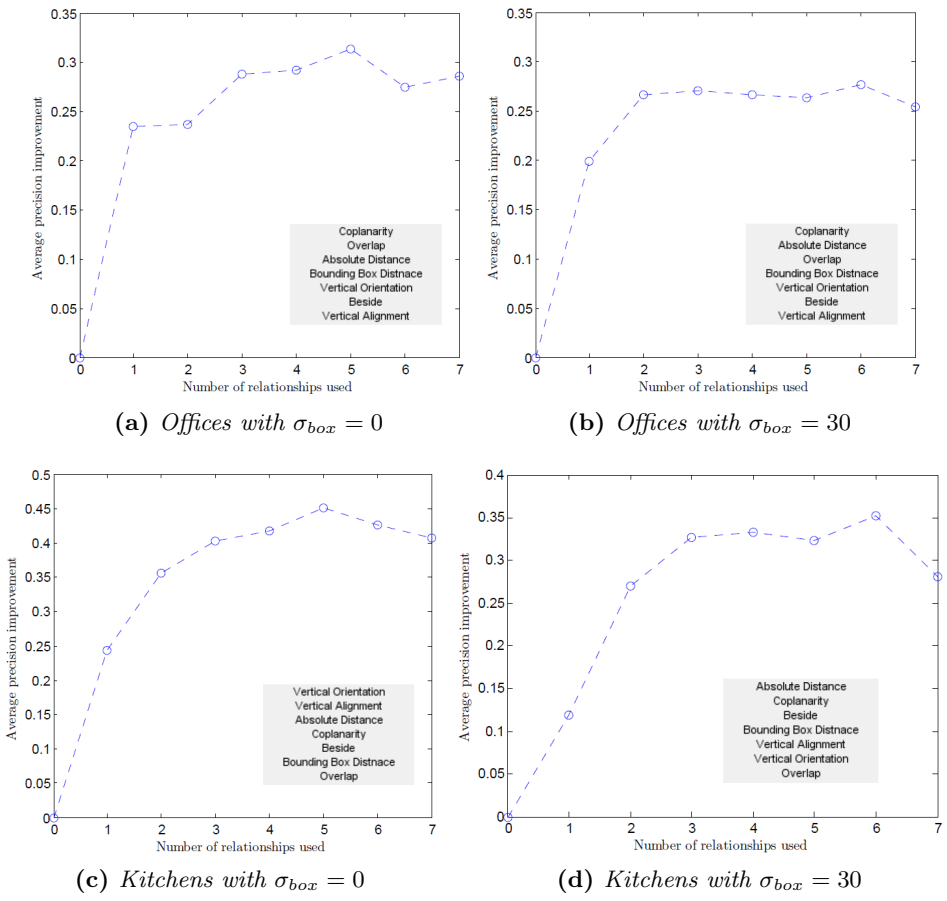
This experiment is designed to rank the effectiveness of the seven spatial relationships we proposed in 7.7. We would like to know which relationships are useful in which scenarios. Is there a smaller set of relationships that work as well or better on all scenes? How does the ranking of useful relationship changes as positional error increases?

We compare the effectiveness of each relationship on both office and kitchen scenes at two levels of location accuracy, good accuracy ($\sigma_{box} = 0cm$) and poor accuracy ($\sigma_{box} = 30cm$), to examine how the best spatial features change when accuracy decreases. We used a greedy approach to ranking the relationships. We performed the experiments by generating 10 versions of each scene and apply each relationship independently. We then took the relationship that led to the highest increase in detection accuracy and repeated the experiment with the combination of that best relationship and each remaining. We repeated this procedure, added a new relationship at each step, until all relationships were used.

Figure 8.17 shows the resulting cumulative benefit from each relationship in both types of scenes. It is evident that there is a plateau effect in most types of scene and little benefit from more than 4 types of relationships. However, it is also evident that the types of relationships best suited to each scenario vary depending on both scene type and localization accuracy. In examining these graphs, it is important to remember that small ranking differences might mean little.

Clearly the effectiveness of no single relationship outperforms all others and a combination of relationships is necessary. There does appear to be a slight overall decrease in performance from using all relationships together. This is likely because the last few added relationships contribute nothing useful for improvement but allow for over-fitted relationship weights to be learned that cause poor performance. With more training data, this would likely not happen as low weights for bad features would be learned. Fortunately, the overall degradation from using all relationships is minimal and likely outweighed by the usefulness of not having to adapt the extracted features to the scene or localization accuracy.

There are no consistently useless relationships and none that clearly dominate. It does appear that the combination of at least one distance-based and one vertical-orientation based relationship is necessary for good results. For distance-based relationships, the absolute distances based on human metrics that we used are more useful than relative bounding box distances but they combine effectively. When localization is poor, fewer features are useful and the combination of absolute distance with coplanarity seems to provide most



(a) *Offices with* $\sigma_{box} = 0$ **(b)** *Offices with* $\sigma_{box} = 30$

(c) *Kitchens with* $\sigma_{box} = 0$ **(d)** *Kitchens with* $\sigma_{box} = 30$

Figure 8.17: *Simulation results that show the cumulative benefit of each of the spatial relationships. The relationships were chosen by greedily selecting the one that led to the greatest improvement at each step. We varied the localization error, controlled by* σ_{box} *and ran the experiment on both offices and kitchens. The order in which the features were applied is shown in the table embedded in each graph.*

of the improvement. With vertical orientation relationships, both vertical orientation and coplanarity were useful when localization is good. Besideness was useful in kitchens, likely because counters and cabinets keep object aligned with each other relative to the walls, something which is not necessarily true in offices. The greater use of overhead spaces and objects embedded into the counters meant that vertical orientation was more useful in kitchens than offices.

Chapter 9

Conclusion

This thesis has examined the effectiveness of 3D spatial relationships for performing or improving object detection and classification. This chapter will discus our conclusions on this subject and the future directions this work could follow.

9.1 Conclusions

Here is a breakdown of our conclusions by chapter:

In Chapter 4 we determined that object detections could provide a good basis for performing scene classification. We believe this is because objects are the basis on which humans create and classify scenes. This scene classification can be used to inform a scene-specific spatial model for improving object detections.

In Chapter 5 we identified the qualitative spatial reasoning literature as a good source of analysis on qualitative spatial relationships. We utilized this research and determined that seven qualitative spatial relationships could cover most of the possible useful binary relationships available when comparing axis-aligned bounding box object detections.

In Chapter 6 we demonstrated that even using simple qualitative spatial relationships it is possible to perform object classification using only information about an object's spatial relationships relative to known objects. This indicates that 3D qualitative spatial relationships are usefully informative to object type and that they can likely be combined with other information, such as detector results, to improve object detection accuracy.

In Chapters 7 and 8 we demonstrated a model for removing false positive 3D detections using a structured support vector machine. We provided a technique for producing 3D detections from 2D detections using a fiducial marker and demonstrated our approach

was successful at significantly improving overall detection rates on real world scenes of both offices and kitchens.

We showed our approach improved on the model it was based upon, that of [Desai et al., 2009], by utilizing a branch and bound tree search to improve both training and inference. The success of our model depended on either having sufficient true positive detections in the training data or having well-localized true positive detections. Our results showed we could effectively handle poor separation between the detection confidence scores of true and false positive detections. Finally, we analyzed the cumulative benefits of the spatial relationships and determined that the most effective spatial relationships depend on both the scene type and localization accuracy. There was no "magic bullet" relationship that was either sufficient on its own or always outperformed others and a mixture of relationships is always useful. However, in general, it was always useful to have a distance-based relationship and one that measured the vertical orientation of the objects. Using all relationships is a good option but a small improvement might be attained by selecting a subset of our relationships appropriate to the localization accuracy and scene type.

Our work also better demonstrated the effectiveness of spatial relationships for improving object detection than that of [Desai et al., 2009] because we removed the factor of co-occurrence between object types. The training and test data they used had 20 classes of object to detect and this broad range of objects came from a correspondingly large number of types of scenes. Therefore, in [Desai et al., 2009] co-occurrence of object types was a very effective way of improving detection rates. For example, if a car was detected in an image, it is unlikely that a chair would also occur and therefore all chair detections can have correspondingly decreased scores. It is unclear how much of the improvement observed in [Desai et al., 2009] was from spatial relationships and how much was from co-occurrence. In our work, we restricted ourselves to single types of scenes and objects that frequently appear in those scenes, so co-occurrence provided little information. This meant our work better demonstrated the effectiveness of spatial relationships by themselves.

In conclusion, we have demonstrated that 3D qualitative spatial relationships can provide an effective basis for improving object detections by removing false positive detections. As long as a robot can identify the scene type, has a mechanism for determining the 3D position of object detections and there is data for constructing a model appropriate for that type of scene, our approach should improve on the overall accuracy of detection. With the increasing prevalence and decreasing price of 3D sensors, we believe that both the 3D detections and data will become increasingly available and that 3D spatial relationships

should be actively explored.

9.2 Future Directions

The use of 3D qualitative spatial relationships for improving object detection is an area of research still in its infancy and we believe there are many directions that could be taken to improve upon our work.

Robot Implementation It would be valuable to demonstrate this work on an actual robot, functioning in real-time in an indoor environment. This goal is mostly a matter of just implementation as there are no major limiting factors preventing this from being achieved. The main missing component is an implementation of an exploration technique which can produce 3D localized detections, which was demonstrated in [Meger et al., 2008]. We opted not to pursue this because implementation of our approach on a robot would have taken a significant amount of time without significantly changing our results.

Improved 3D Object Detection Localization One improvement we believe would be highly effective at improving on our work would be a better technique for determining the position, size and extent of the 3D detections and a representation that could utilize this data. Our approach was limited by the fact that we relied on only images of the scene and often only had two or three detections on which to determine the object position. As such, we were limited in how much information we could reliably determine about the object's position and opted to only determine the position of the centroid and infer the size of the object detection from the average sizes previously observed in the training data.

We believe that if we were using an actual robot in the environment, equipped with a camera to determine the visual detection position and a scanning 3D range finder to determine the size and extend of the object, there would be several advantages. Firstly, we have shown that better localization significantly improves upon our overall accuracy. A 3D range finder should allow the robot to much better approximate the location of many objects. Secondly, by moving from axis-aligned bounding boxes to a representation of size and shape of the object, new spatial relationships such as size comparison and relative topology become available.

Non-scene Specific Model Our final model for improving object detection accuracy was scene-specific and, in this thesis, we have not explored the options for using a non-

scene specific model. Our motivation for using a scene specific model was two fold. Firstly, our data sets consisted only of individual scenes and there were no examples which contained both offices and kitchens. Secondly, it simplified the overall learning problem by only learning relationships between objects found in the same environments. We decided this simplification was necessary because of the limited available training data and demonstrated it was reasonable by showing that scenes could be classified with good accuracy based on initial object detections.

However, buildings are not simply collections of scenes and there are many environments that don't have an obvious scene label. We believe that our approach would scale up to a scene independent model if we have more available training data. Our evidence for this is that the model in [Desai et al., 2009] was not scene dependent and was able to handle learning a much larger number of object type pairs than we attempted. One advantage of the scene dependent model is obviously that it can learn relationships that are scene dependent (e.g., coffee cups in kitchens are found in different locations than coffee cups in offices). So future work should explore ways of mediating between scene dependent and independent models and assessing which is appropriate for a given situation.

Expanded Scene and Object Categories It would be useful to demonstrate this work on a larger collection of scene types and with more object types. Specifically, it would be interesting to explore less structured scenes like living rooms or large scenes like shops. More scene types would give us a better insight into the value of different types of spatial relationships and different representations of object detections. More object types, if they can be detected, should hopefully lead to improved results as we have demonstrated that having more true positive detections improves our overall results.

Learn the Qualitative Spatial Relationships This thesis used spatial relationships that were derived from the Qualitative Spatial Reasoning literature and other work on context aided computer vision. As such, the spatial relationships were created from human expert experience rather than learned directly from observations of the scenes. An interesting future direction for this work would be to learn qualitative spatial relationships from quantitative spatial information about scenes. We suspect that learning qualitative spatial relationships would require significantly more 3D scene data than we were able to produce for this thesis.

Bibliography

M. Aiello and J. van Benthem. *Logical Patterns in Space*. ILLC scientific publications. Institute for Logic, Language and Computation, 1999.

M. A. Aizerman, E. A. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. In *Automation and Remote Control*, pages 821–837, 1964.

J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–943, 1983.

S. Andrews, T. Hofmann, and I. Tsochantaridis. Multiple instance learning with generalized support vector machines. In *American Association for Artificial Intelligence*, pages 943–944, 2002.

D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learning of Markov random fields for segmentation of 3D range data. In *Computer Vision and Pattern Recognition*, 2005.

C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.

S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

Bethesda Softworks. Elder scrolls 4: Oblivion. [DVD-ROM], 2006.

A. Bosch, X. Muñoz, and R. Martí. Review: Which is the best way to organize/classify images by content? *Image and Vision Computing*, 25(6):778–791, 2007.

L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *European Conference on Computer Vision*, 2010.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

S. B. Carolina Galleguillos, Brian McFee and G. R. G. Lanckriet. Multi-class object localization by combining local contextual interactions. In *Computer Vision and Pattern Recognition*, pages 113–120, 2010.

E. Clementini and P. D. Felice. Approximate topological relations. *International Journal of Approximate Reasoning*, 16(2):173–204, 1997.

A. G. Cohn and S. M. Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*, 46:1–29, 2001.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, Mar. 2002.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, pages 886–893, 2005.

T. de Laguna. Point, line, and surface, as sets of solids. *The Journal of Philosophy*, 19: 449–461, 1922.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, pages 248–255, 2009.

C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *International Conference on Computer Vision*, pages 229–236, 2009.

T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.

S. K. Divvala, D. Hoiem, J. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. In *Computer Vision and Pattern Recognition*, pages 1271–1278, 2009.

M. Duckham, J. Lingham, K. T. Mason, and M. F. Worboys. Qualitative reasoning about consistency in geographic information. *Information Sciences*, 176(6):601–627, 2006.

E. F. Ersi and J. K. Tsotsos. Visual place categorization in indoor environments. In *Canadian Conference on Computer and Robot Vision*, pages 448–453, 2012.

T. Escrig and F. Toledo. *Qualitative Spatial Reasoning Theory and Practice: Theory and Practice: Application to Robot Navigation*. Frontiers in Artificial Intelligence and Applications, 47. IOS PressInc, 1998.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

P. F. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition*, 2008.

R. Fergus, F. Li, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *International Conference on Computer Vision*, 2005.

M. Fiala. Artag, a fiducial marker system using digital techniques. In *Computer Vision and Pattern Recognition*, pages 590–596, 2005.

D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic. People watching: Human actions as a cue for single-view geometry. In *European Conference on Computer Vision*, pages 732–745, 2012.

A. U. Frank. Qualitative spatial reasoning about cardinal directions. In *International Joint Conference on Artificial Intelligence*, pages 157–167, 1991.

J. Freeman. The modeling of spatial relations. *Computer Graphics and Image Processing*, 4:156–171, 1975.

C. Freksa. Using orientation information for qualitative spatial reasoning. In *SpatioTemporal Reasoning*, pages 162–178, 1992.

Y. Freund. An adaptive version of the boost by majority algorithm. *Machine Learning*, 43 (3):293–318, 2001.

Y. Freund and L. Mason. The alternating decision tree learning algorithm. In *International Conference on Machine Learning*, pages 124–133, 1999.

Y. Freund and R. E. Schapire. A brief introduction to boosting. In *International Joint Conference on Artificial Intelligence*, pages 1401–1406, 1999.

J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2):337–374, 2000.

C. Galleguillos and S. Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722, 2010.

C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *Computer Vision and Pattern Recognition*, 2008.

S. Gottschalk, M. C. Lin, and D. Manocha. OBBTree: A hierarchical structure for rapid interference detection. In *SIGGRAPH*, pages 171–180, 1996.

K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 8:725–760, 2007.

J. Hays and A. A. Efros. Im2gps: estimating geographic information from a single image. In *Computer Vision and Pattern Recognition*, pages 1–8, 2008.

V. Hedau, D. Hoiem, and D. A. Forsyth. Recovering the spatial layout of cluttered rooms. In *International Conference on Computer Vision*, pages 1849–1856, 2009.

S. Helmer and D. Lowe. Using stereo for object recognition. In *International Conference of Robotics and Automation*, pages 3121–3127, 2010.

D. Hernndez and K. Zimmermann. *Qualitative Representation of Spatial Knowledge*. Lecture Notes in Artificial Intelligence. Springer, 1994.

D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In *Computer Vision and Pattern Recognition*, pages 2137–2144, 2006.

D. Hoiem, C. Rother, and J. Winn. 3D LayoutCRF for multi-view object class recognition and segmentation. In *Computer Vision and Pattern Recognition*, June 2007.

X. Jin and J. Han. Quality threshold clustering. In *Encyclopedia of Machine Learning*, page 820. Springer, 2010.

A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3D Mesh Segmentation and Labeling. *ACM Transactions on Graphics*, 29(3), 2010.

A. Kreutzmann, K. Terzić, and B. Neumann. Context-aware classification for incremental scene interpretation. In *Workshop on Use of Context in Vision Processing*, pages 1–6, 2009.

B. Kröse, O. Booij, and Z. Zivkovic. A geometrically constrained image similarity measure for visual mapping, localization and navigation. In *European Conference on Mobile Robots*, pages 168 – 174, Freiburg, Germany, 2007.

H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, 1951.

B. Kuipers, R. Browning, B. Gribble, M. Hewett, and E. Remolina. The spatial semantic hierarchy. *Artificial Intelligence*, 119:191–233, 2000.

J.-F. Lalonde, S. G. Narasimhan, and A. A. Efros. What does the sky tell us about the camera. In *European Conference of Computer Vision*, pages 354–367, 2008.

S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.

D. C. Lee, A. Gupta, M. Hebert, and T. Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *Advances in Neural Information Processing Systems*, pages 1288–1296, 2010.

F.-F. Li and L.-J. Li. What, where and who? telling the story of an image by activity classification, scene recognition and object categorization. In *Computer Vision: Detection, Recognition and Reconstruction*, volume 285 of *Studies in Computational Intelligence*, pages 157–171. Springer, 2010.

F.-F. Li and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition*, pages 524–531, 2005.

L.-J. Li, H. Su, Y. Lim, and F.-F. Li. Objects as attributes for scene classification. In *European Conference of Computer Vision, Workshop on Parts and Attributes*, pages 57–69, 2010.

S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer Publishing Company, Incorporated, 3rd edition, 2009.

C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. In *European Conference on Computer Vision*, pages 28–42, 2008.

D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

M. Mavrovouniotis and G. Stephanopoulos. Formal order-of-magnitude reasoning in process engineering. *Readings in qualitative reasoning about physical systems*, pages 323–336, 1990.

M. L. Mavrovouniotis and G. Stephanopoulos. Order-of-magnitude reasoning with O[M]. *AI in Engineering*, 4(3):106–114, 1989.

D. Meger, P.-E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. J. Little, and D. G. Lowe. Curious George: An attentive semantic robot. *Robotics and Autonomous Systems*, 56(6):503–511, June 2008.

D. Meger, C. Wojek, B. Schiele, and J. J. Little. Explicit occlusion reasoning for 3D object detection. In *British Machine Vision Conference*, 2011.

J. Modayil and B. Kuipers. Bootstrap learning for object discovery. In *International Conference on Intelligent Robots and Systems*, 2004.

D. R. Montello. Scale and multiple psychologies of space. In *Conference On Spatial Information Theory*, pages 312–321, 1993.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *International Joint Conference on Artificial Intelligence*, 2003.

S. G. Narasimhan and S. K. Nayar. Vision and the atmosphere. *International Journal of Computer Vision*, 48(3):233–254, 2002.

A. Neubeck and L. Van Gool. Efficient non-maximum suppression. In *Internation Conference on Pattern Recognition*, pages 850–855, 2006.

B. Neumann. Bayesian compositional hierarchies - a probabilistic structure for scene interpretation. In *Logic and Probability for Scene Interpretation*, 2008.

B. Neumann and R. Möller. On scene interpretation with description logics. *Image and Vision Computing, Special Issue on Cognitive Vision*, 26(1):82–101, 2007.

A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen. A discriminative approach to robust visual place recognition. In *International Conference on Intelligent Robots and Systems*, pages 3829–3836, 2006.

A. Pronobis, O. M. Mozos, and B. Caputo. SVM-based discriminative accumulation scheme for place recognition. In *International Conference on Robotics and Automation*, pages 522–529, 2008.

A. Pronobis, O. M. Mozos, B. Caputo, and P. Jensfelt. Multi-modal semantic place classification. *The International Journal of Robotics Research*, 29:298–320, 2010.

A. Quattoni and A. B. Torralba. Recognizing indoor scenes. In *Computer Vision and Pattern Recognition*, pages 413–420, 2009.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Mateo, CA, 1993.

D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *Knowledge Representation*, pages 165–176, 1992.

A. Ranganathan and F. Dellaert. Semantic modeling of places using objects. In *Robotics: Science and Systems*, 2007.

A. Ranganathan and J. Lim. Visual place categorization in maps. In *International Conference on Intelligent Robots and Systems*, pages 3982–3989, 2011.

J. Renz. *Qualitative spatial reasoning with topological information*. Lecture Notes in Computer Science. Springer-Verlag, 2002.

R. Rimey and C. Brown. Control of selective perception using Bayes nets and decision theory. *International Journal of Computer Vision*, 12:173–207, 1994.

F. Roberts and P. Suppes. Some problems in the geometry of visual perception. *Synthese*, 17(1):173–201, 1967.

B. Russell, A. Torralba, K. Murphy, and W. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77:157–173, 2008.

B. C. Russell, A. Torralba, C. Liu, R. Fergus, and W. T. Freeman. Object recognition by scene alignment. In *Advances in Neural Information Processing Systems*, 2007.

R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39:135–168, 2000.

C. Schlieder. Representing visible locations for qualitative navigation. In *Qualitative reasoning and decision technologies*, pages 523–532. CIMNE, 1993.

J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1270–1281, July 2008.

J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009.

B. Siciliano and O. Khatib. *Springer Handbook of Robotics*. Springer, Berlin, 2008.

N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference of Computer Vision*, pages 746–760, 2012.

I. Simon and S. M. Seitz. Scene segmentation using the wisdom of crowds. In *European Conference of Computer Vision*, pages 541–553, 2008.

P. Simons. *Parts: A Study in Ontology*. Routledge, 1987.

A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

T. Southey and J. J. Little. Object discovery through motion, appearance and shape. In *American Association for Artificial Intelligence, Workshop on Cognitive Robotics*. AAAI Press, 2006.

T. Southey and J. J. Little. 3d spatial relationships for improving object detection. In *International Conference on Robots and Automation*, 2012.

T. Spexard, S. Li, B. Wrede, J. Fritsch, G. Sagerer, O. Booij, Z. Zivkovic, B. Terwijn, and B. Kröse. Biron, where are you? - enabling a robot to learn new places in a real home environment by integrating spoken dialog and visual localization. In *International Conference on Intelligent Robots and Systems*, 2006.

T. Strat. Employing contextual information in computer vision. In *DARPA Image Understanding Workshop*, pages 217–229, 1993.

A. Swadzba and S. Wachsmuth. Aligned scene modeling of a robot's vista space - an evaluation. In *AAAI Workshop on Language - Action Tools for Cognitive Artificial Agents: Integrating Vision, Action and Language*, pages 30–35, 2011.

B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *Advances in Neural Information Processing Systems*. MIT Press, 2003.

K. Terzić and B. Neumann. Context-based probabilistic scene interpretation. In *Internation Conference on Artificial Intelligence in Theory and Practice*, pages 155–164, 2010.

A. R. Tilley and S. B. Wilcox. *The measure of man and woman: human factors in design*. Wiley, New York, 2002.

A. Torralba, K. Murphy, W. T. Freeman, and M. Rubin. Context-based vision system for place and object recognition. In *International Conference on Computer Vision*, pages 273–280, 2003.

A. Torralba, K. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems*, 2004.

A. Torralba, K. P. Murphy, and W. T. Freeman. Using the forest to see the trees: exploiting context for visual object detection and localization. *Communications of the ACM*, 53(3): 107–114, 2010.

G. Tsai, C. Xu, J. Liu, and B. Kuipers. Real-time indoor scene understanding using Bayesian filtering with motion cues. In *International Conference on Computer Vision*, pages 121–128, 2011.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine learning*, 2004.

S. Vasudevan and R. Siegwart. A Bayesian space conceptualization and place classification for semantic maps in mobile robotics. *Robotics and Autonomous Systems*, 56(6):522–537, June 2008.

A. Vedaldi. A MATLAB wrapper of SVM$^{\text{struct}}$. `http://www.vlfeat.org/~vedaldi/code/svm-struct-matlab.html`, 2011.

L. Vieu. Spatial representation and reasoning in AI. In *Spatial and Temporal Reasoning*, pages 5–41, 1997.

P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

P. Viswanathan, D. Meger, T. Southey, J. J. Little, and A. Mackworth. Automated spatial-semantic modeling with applications to place labeling and informed search. In *Canadian Conference on Computer and Robot Vision*, pages 284–291, 2009.

P. Viswanathan, T. Southey, J. J. Little, and A. Mackworth. Place classification using visual object categorization and global information. In *Canadian Conference on Computer and Robot Vision*, pages 1–7, 2011.

J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *Computer Vision and Pattern Recognition*, pages 37–44, 2006.

J. Wu, H. I. Christensen, and J. M. Rehg. Visual place categorization: Problem, dataset, and algorithm. In *International Conference on Intelligent Robots and Systems*, pages 4763–4770, 2009.

C. Xu and B. Kuipers. Towards the object semantic hierarchy. In *International Conference on Development and Learning*, 2010.

Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *International Conference on Machine Learning*, pages 412–420, 1997.

J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.