

2013

## Automated Semantic Content Extraction from Images

Mahdi Arab Khazaeli

*Louisiana State University and Agricultural and Mechanical College*

Follow this and additional works at: [https://digitalcommons.lsu.edu/gradschool\\_dissertations](https://digitalcommons.lsu.edu/gradschool_dissertations)



Part of the [Engineering Science and Materials Commons](#)

---

### Recommended Citation

Arab Khazaeli, Mahdi, "Automated Semantic Content Extraction from Images" (2013). *LSU Doctoral Dissertations*. 2697.

[https://digitalcommons.lsu.edu/gradschool\\_dissertations/2697](https://digitalcommons.lsu.edu/gradschool_dissertations/2697)

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Doctoral Dissertations by an authorized graduate school editor of LSU Digital Commons. For more information, please contact [gradetd@lsu.edu](mailto:gradetd@lsu.edu).

# AUTOMATED SEMANTIC CONTENT EXTRACTION FROM IMAGES

A Dissertation

Submitted to the Graduate Faculty of the  
Louisiana State University and  
Agricultural and Mechanical College  
in partial fulfillment of the requirements for the degree of  
Doctor of Philosophy

in

The Interdepartment Program in  
Engineering Science

by

Mehdi A. Khazaeli

B.S., Isfahan University of Technology, 2006

M.S., University of Liverpool, 2009

August, 2013

## **ACKNOWLEDGEMENTS**

I express my sincere appreciation to my advisor Dr. Gerald M. Knapp for his constant guidance, help and constructive criticism throughout my program of study. His depth of knowledge and enthusiasm helped me learn and grow in an emerging area of research at the cross section between industrial engineering and computer science which I truly enjoy. I am grateful for having him as an advisor.

I also thank my committee members, Dr.'s Bahadir Gunturk, Laura Ikuma and Warren Waggenspack, for their guidance and support during my studies. Their patience is much appreciated. Thanks to Dr. Gunturk for giving me much inspiration and sharing his rich and valuable knowledge in image processing and pattern recognition.

This work would not be possible without the help and support of Mr. Mark, Mrs. Naomi, Matthew and Amy Valliollahi for allowing me to be part of their loving family, teaching me valuable lessons in life and celebrating all my accomplishments with me. My gratitude to Dr. Merrikh Ramazanian, Dr. Mahmood Sabahi and Dr. Fereydoun Aghazadeh, their generosity and kindness will never be forgotten.

To my fellow graduate students, Dr. Ricardo Calix, Sri Abhishikth Mallepudi and Soha Khazaeli, I extend my appreciation. I wish you all the best in your careers. Special thanks go to Dr.'s Laura Ikuma, Carol Freidland and Peter Chen for their help with financial support during the summers.

The most substantial contributions to my entire education are, of course, from my parents, siblings Ali, Roja, Soha, my in-laws and friends who have been supportive and helpful throughout my life.

Finally and most importantly, without the multiple sacrifices made by my best colleague, friend and wife, Leili, this work would have been impossible. Her support, encouragement and unwavering love were undeniably the bedrock upon which the past ten years of my life have been built.

# TABLE OF CONTENTS

ACKNOWLEDGEMENTS .....	ii
TABLE OF CONTENTS.....	iv
LIST OF TABLES .....	vi
LIST OF FIGURES .....	vii
ABSTRACT.....	ix
CHAPTER 1 – PROBLEM STATEMENT.....	1
1.1 Objectives.....	8
CHAPTER 2- LITERATURE REVIEW .....	10
2.1 Content-Based Image Retrieval Systems .....	10
2.1.1 Review of Existing CBIR Systems.....	12
2.2 Multimedia Analysis Using Kinect .....	14
2.3 Features and Feature Selection.....	18
2.3.1 Color .....	20
2.3.2 Texture.....	21
2.3.3 Shape .....	21
2.3.4 Depth .....	22
2.4 Overview of Machine Learning Approaches .....	23
2.4.1 Supervised Methods .....	23
2.4.2 Unsupervised Methods .....	25
2.5 Image Segmentation.....	26
2.6 Surface Detection .....	29
2.7 Scene Detection.....	30
2.8 Object Recognition.....	31
2.9 Commonsense Knowledge.....	35
2.10 Corpora.....	36
CHAPTER 3 - AUTOMATED SEMANTIC CONTENT EXTRACTION APPROACH.....	38
3.1 Tools.....	40
3.2 Corpus .....	40
CHAPTER 4- SEGMENTATION.....	43
4.1 Camera Calibration .....	44
4.2 Pixel Features .....	47
4.2.1 Capturing Depth and RGB .....	48
4.2.2 Projecting Depth on RGB Plane .....	50
4.2.3 Finding XYZ in World Coordinates.....	51
4.2.4 Computing Normals .....	52
4.2.5 Finding Vanishing Points .....	53

4.2.6 Rotating the Scene and Normalizing Features .....	54
4.2.7 Texture Features .....	55
4.3 Surface.....	57
4.4 Graph Based Segmentation .....	61
4.5 Clustering .....	63
4.6 Post-processing of Object Segmentation.....	65
4.7 Time Analysis .....	66
CHAPTER 5- OBJECT RECOGNITION .....	67
5.1 Finding Room Surfaces .....	67
5.2 Generating Feature Vector for Each Object.....	69
5.3 Object Feature Analysis .....	71
CHAPTER 6- SCENE IDENTIFICATION .....	76
6.1 Global Feature Vector .....	77
6.2 Scene Feature Analysis .....	81
CHAPTER 7- EVIDENCE FUSION MODEL .....	83
7.1 Scene Likelihood.....	83
7.2 Object Likelihood.....	86
7.3 ConceptNet Similarity .....	88
7.4 Fusion Method and Results .....	88
7.5 Text Generation.....	91
CHAPTER 8- CONCLUSION AND FUTURE WORK.....	94
8. 1 Recommendations for Future Work.....	95
REFERENCES .....	97
VITA.....	106

## LIST OF TABLES

Table 1: Machine learning techniques (Calix 2011).....	24
Table 2: ConceptNet semantic relations .....	36
Table 3: Example of object dataset consisting single object images in $427 \times 561$ pixels of size on a white background .....	42
Table 4: Calibration parameters for depth and color cameras .....	48
Table 5: Time Breakdown .....	66
Table 6: Feature vector for object recognition.....	71
Table 7: Classification results for object recognition .....	73
Table 8: Confusion matrix for object recognition.....	73
Table 9: Texture features ranking .....	75
Table 10: Classification results for scene identification .....	82
Table 11: Confusion matrix .....	82
Table 12: Classification Results for test Images .....	85
Table 13: Performance for each class .....	85
Table 14: Test images with scene likelihoods .....	86
Table 15: Test images with object likelihoods .....	87
Table 16: ConceptNet activation measure .....	88
Table 17: Object recognition result for a sample image .....	90

## LIST OF FIGURES

Figure 1: Diagram for content-based image retrieval system (Siu and Zhang 2003).....	11
Figure 2: The Kinect components .....	15
Figure 3: ConceptNet selected output for "cat" .....	36
Figure 4: Diagram of the framework .....	39
Figure 5: Image hierarchy .....	39
Figure 6: Segmentation engine .....	43
Figure 7: Calibrate the Kinect using chessboard .....	45
Figure 8: Left-Calibration of color camera; Right-Calibration of infrared camera .....	46
Figure 9: Showing points with no depth data .....	50
Figure 10: Projecting depth on color image.....	51
Figure 11: Showing XYZ in world coordinates of chair in front of wall .....	52
Figure 12: Detected line segments .....	54
Figure 13: Rotation of the camera and relative rotation between the cameras .....	55
Figure 14: Surface detection for a sample image.....	59
Figure 15: Surface detection for a sample image.....	60
Figure 16: Graph-based segmentation .....	62
Figure 17: Segmentation result of an bathroom image .....	64
Figure 18: Segmentation of a classroom image .....	64
Figure 19: Tiny polygons created in segment boundaries .....	65
Figure 20: Segmentation result after applying median filtering (right image) .....	66
Figure 21: A classroom with its floor and wall specified .....	68



Figure 22: A bathroom with specified structures.....	69
Figure 23: Histogram of gradient.....	70
Figure 24: Chi square feature ranking showing the top 20 features .....	74
Figure 25: HOG representation for corridor .....	80
Figure 26: Visualizing the computation of object label.....	89
Figure 27: Testing the performance of the system for occlusion and illumination challenges ....	91
Figure 28: Assertions extracted from ConceptNet for ‘Chair’ and ‘Computer’ .....	92
Figure 29: XML format enrichment for an image .....	93

## **ABSTRACT**

In this study, an automatic semantic segmentation and object recognition methodology is implemented which bridges the semantic gap between low level features of image content and high level conceptual meaning. Semantically understanding an image is essential in modeling autonomous robots, targeting customers in marketing or reverse engineering of building information modeling in the construction industry. To achieve an understanding of a room from a single image we proposed a new object recognition framework which has four major components: segmentation, scene detection, conceptual cueing and object recognition.

The new segmentation methodology developed in this research extends Felzenswalb's cost function to include new surface index and depth features as well as color, texture and normal features to overcome issues of occlusion and shadowing commonly found in images. Adding depth allows capturing new features for object recognition stage to achieve high accuracy compared to the current state of the art. The goal was to develop an approach to capture and label perceptually important regions which often reflect global representation and understanding of the image.

We developed a system by using contextual and common sense information for improving object recognition and scene detection, and fused the information from scene and objects to reduce the level of uncertainty. This study in addition to improving segmentation, scene detection and object recognition, can be used in applications that require physical parsing of the image into objects, surfaces and their relations. The applications include robotics, social networking, intelligence and anti-terrorism efforts, criminal investigations and security, marketing, and building information modeling in the construction industry. In this dissertation a structural

framework (ontology) is developed that generates text descriptions based on understanding of objects, structures and the attributes of an image.

**Keywords:** object recognition, contextual cueing, scene detection, segmentation, commonsense knowledge and machine learning.

## **CHAPTER 1 – PROBLEM STATEMENT**

The rapid growth in availability of high-quality digital cameras and video recorders, and the development of easy-to-use large-scale internet archives such as Facebook and YouTube for sharing photos and videos, has resulted in enormous collections of unstructured or loosely structured images and videos. The photo-hosting Website, Flickr, uploaded its six-billionth picture on August 5, 2011 and Pingdom, a Swedish company that monitors internet performance, estimated that photos are being added to Facebook at a rate of 109.5 billion per year which translates to 250TB of additional storage consumed weekly.

Semantic understanding of image scenes and their component objects is an important current research challenge. While face recognition has seen significant improvements in recent years, the ability to identify other objects in scenes is still quite primitive. There are many important and potentially lucrative applications of such capabilities, such as:

- Reverse Engineering. Having a person or robot navigate an existing building with an imaging system to reverse engineer the structure and its contents (such as furnishings) into a 3D building information model (BIM), for use in architectural renovations, fire/safety analysis and planning, police/military action planning, and criminal investigations.
- Marketing applications. Marketing companies may be interested in mining pictures in social media to determine the types of décor and furniture people have in their homes/rooms for advertising purposes.

- Computer-based image retrieval (CBIR). Users of large image databases may want to query for pictures containing specific items, such as bookcases. Currently, users can only access such images if they have been tagged in some way (text captions, XML tags).

There are significant challenges for object recognition in images. These include:

- Varying weather and lighting conditions, and exposure settings, which can change coloration and introduce or alter color gradients, shadows, and glare.
- Picture focus.
- Picture resolution.
- Viewpoint and zoom variations. Pictures of the same scene can be taken from different angles, heights, distances, and zoom settings.
- Occlusions. Color-only pictures are a 2D representation of a 3D world. Objects in the foreground occlude objects further back, and may "split" background objects into multiple segments. Determining which segments go together is a non-trivial computational task.
- Shape changes. Non-rigid objects, such as people and bean bag chairs, can change pose and shape. Even rigid objects such as chairs can often be changed in configuration (height, tilt).
- Intra object class variability. There are many variations in shape, material, and texture for a single object type. For instance, a desk can be made from various combinations of different woods, metals, and glass; can come in different colors and textures; can have different types of supports (two, four, and even zero legs for cantilevered desks); and can have different configurations of drawers and keyboard trays.

This research has focused on advancing the state of the art in automated object and scene identification in interior residential and office spaces. In particular, we have focused on how new inexpensive technologies providing depth information can be used to resolve some of the issues above; how the typical geometric structural components of indoor scenes such as floors, walls, and ceilings can be utilized to help in separating objects; how identification of the scene itself (for instance, bathroom versus classroom) can be used to improve object identification; and extraction of additional semantic relations between objects in the scene.

Since the 1990's, a considerable amount of image analysis research has focused on extending information retrieval (IR) techniques used in text retrieval to the area of image retrieval. In Content-Based Image Retrieval (CBIR) systems (Porkaew et al. 1999; Vailaya et al. 2001), image processing techniques are used to extract relatively low level visual features such as color, texture and shape from images. Images or image regions are represented as a collection, or vector of values for the visual features extracted. A user formulates a query by providing examples of images similar to the ones he/she wishes to retrieve. The system converts this into an internal representation of a query, based on features extracted from input images. Retrieval is performed based on computing similarity between images in the archive and query representation in the feature space, and the results are ranked based on the computed similarity values.

CBIR and image content identification still has not had a substantial impact on real world applications. Current web-based image search engines rely primarily on metadata information accompanying images, such as caption titles, keywords, tags, or descriptions associated with the image to provide semantic content (Crandall et al. 2009). The main limitations of existing CBIR

systems include lack of adequate unique correlations between low level features and high level semantic concepts, limitations on accuracy of region segmentation algorithms, and usability disconnect between how users want to query images and how CBIR requires queries. In the past decade, CBIR and image content identification systems using image-recognition technologies were applied to applications in industrial automation, robotics, biomedicine, social apps and more fields and proved to be reliable. Face-recognition systems are being used in biometric<sup>1</sup> authentication and crime prevention (Chellappa et al. 1995). In medicine, automatic image-based detection of tumor cells is being used for medical diagnosing (Shyu et al. 1999). In landmark recognition, metadata such as GPS-based tags are used. These geo-tags are used to learn what the pictures are about (Crandall 2009).

Despite all the work done in CBIR and image content identification, the problem of object identification is still relatively unsolved.

While expensive depth-finding systems (such as laser scanners) have been around for some time, recently a number of very capable low cost devices have become commercially available which can provide high quality features containing both depth and color information. Hartley and Zisserman demonstrated the positive impact of depth information in computer vision (Hartley and Zisserman 2000). Different approaches have been developed to increase the accuracy of object recognition and segmentation. In some of the studies, existence of a specific object is assumed, and some studies require interactions with users to identify presence of objects. In this research a combination of local and global features is studied and an automated model for image

---

<sup>1</sup> Biometrics deal with the automated recognition of individuals based on *physiological* (e.g. fingerprint, face, iris) and *behavioral* (e.g. gait, signature, keystroke) characteristics.

retrieval is developed. This model can be implemented on any kind of objects (rigid or deformable) and no requirements are needed for existence of specific objects (Weerasinghe et al. 2010; Eguchi and Thompson 2012; Grabner et al. 2011).

In this research, Microsoft Kinect is utilized, which was commercially introduced in 2009. Features of the Kinect include color image retrieval, depth image retrieval, human body recognition, skeleton joint tracking, and a multi-array microphone (Microsoft Kinect 2013).

In this research, improvements were developed in four major component areas of the object identification problem: segmentation of the image, object recognition of the main segments, scene detection of each image, and contextual cueing. Each component is explained briefly in the following.

Segmentation identifies the parts of an image that are likely to correspond to individual objects. Much research and many algorithms have been presented such as Blobworld (Carson et al. 2000), counter models (Chen et al. 2010), region growing (Deng and Manjunath 2001) and graph based segmentation (Shi and Malik 2000). A wide range of vision problems such as object recognition and motion estimation require the image to be segmented into regions for further analyzing, therefore segmentation is a fundamental task for image processing. Our goal is to develop an approach to capture perceptually important regions which often reflect global representation of the image. State of art methods generally fail to capture important non-local properties of an image (Felzenszwalb and Huttenlocher 2004). The implemented method in this research captures a combination of pixel level and global features and generates a new feature vector for segmentation process.



Object recognition involves identifying the presence of a particular object in an image. Recent techniques that show promising results in identifying objects include template matching (Dufour et al. 2002), detectors using histogram of oriented gradients feature descriptors (Dalal and Triggs 2005), deformable templates and part based models (Fischler and Elschlager 1973). The problem of detecting objects using these models is that they need to rely on local search techniques for performing detection. Recent work done in this area shows that using object detection systems based on mixtures of multi-scale deformable part models achieved 20-30% average precision (AP) on the PASCAL dataset in recognizing 20 different object classes (Desai et al. 2009; Sadeghi and Farhadi 2011; Dalal and Triggs 2005). The reason for the low accuracy is that 2D images are not as informative as real world 3D scenes. By converting to 2D images some features are lost and the remaining features are not adequate to retrieve the entire information from the picture. Consequently, researchers are looking for ways to add more features using new hardware capabilities. These technologies include laser scanning and multi-view geometry for capturing depth, motion and spatial features.

Many shape processing theories invoke contour features (angles and curves) and use salient edges (Canny 1986), corner detection algorithms (Trujillo and Olague 2008) and templates which provide a complementary description of image structures in terms of regions to construct more complex representations. Once features have been detected, a local image patch around the feature can be extracted. The result is known as a feature descriptor or feature vector. The Hough transform approach, local histogram of oriented gradient (HOG) approach and scale invariant feature transform (SIFT) are used for extracting feature description. The Hough transform technique finds imperfect instances of objects within a certain class of shapes by a voting procedure (Hough 1962). HOG is based on a technique which counts occurrences of gradient

orientation in localized portions of an image (Dalal and Triggs 2005). The SIFT feature descriptor transforms an image into feature vectors which are invariant to uniform scaling, orientation and affine distortion (Lowe 1999). By adding depth and its derivations, new sets of features are generated to increase the accuracy of object recognition.

For scene recognition, the current state of art approaches represent scenes with global features, measuring color histogram parameters on different orientations and scales, and taking into account general information about images. Recognizing objects such as sky, snow, rock, water, sand and etc. can provide important context information for an image. In contrast to extracting global features, researchers use semantic information as attributes of images for scene identification (Li et al. 2009; Oliva and Torralba 2006). Scene classification is a subset of the image understanding problem, and can be used to improve other image understanding tasks (Loschky et al. 2007).

One of the main problems of object recognition is matching a representation of the target object with the available image features, while rejecting the background features. However, in the real world, other objects in a scene are a rich source of information that can serve to help recognition and detection of objects. Scene can provide an extremely effective source of information for contextual cueing. In this study scene information is fused with local features of a segment to predict the identity of the object.

An object identification system can also identify additional properties of objects (such as color or texture), and relations between the objects in an image. There have been attempts to generate sentences or graph models from visual data. Some models use AND-OR graphs (Gupta and

Davis 2008) while others use metadata for describing scenes or generating narrative sentences (Yao et al. 2010; Farhadi et al. 2010).

Despite decades of research in image analysis, a substantial gap still exists between the primarily low level image features currently extractable and the mapping of these features to higher level semantic concepts (e.g. dog, person, building, and car). A new system is developed that uses contextual and common sense information for improving the object recognition and scene detection. Information of scene and objects were fused together to reduce the level of uncertainty. This study in addition to improving segmentation, scene detection and object recognition can be used in different applications. The applications include robotics, social networking, intelligence and anti-terrorism efforts, criminal investigations and security, marketing, and linking feature descriptors to a BIM to support building object recognition.

## **1.1 Objectives**

The major objectives of this study are:

- Creating an annotated dataset for training and testing purposes which consist of images with labeled scene and objects.
- Developing a new framework for segmenting an image by integrating several existing bottom-up and top-down methods.
- Developing a new set of features for object recognition.
- Developing a new scene detection model by integrating local gradients and global color and texture features.

- Developing an evidence fusion model combining scene detection, object detection findings and commonsense knowledge.
- Evaluating performance of the model against the annotated corpus.

## **CHAPTER 2- LITERATURE REVIEW**

In this chapter, content-based image retrieval systems are introduced and their characteristics are studied. Kinect is also introduced and the advantages of using this sensor are explained. The applications and developments in the area of object recognition and content based image retrieval are discussed. Then a broad overview of existing research related to the topics of segmentation, object recognition, scene detection and contextual cueing is provided. Extracting feature vectors for segmentation and object recognition are the preliminary approaches in this dissertation work; therefore these methods are presented and discussed in more detail. Statistical and probabilistic machine learning methods are mentioned for understanding the main machine learning techniques.

### **2.1 Content-Based Image Retrieval Systems**

Image retrieval approaches were designed based on the information retrieval techniques applied to the retrieval of text documents. Barnard et al. named two applications of methods that link text and images: Illustration, where one finds pictures suggested by text; and annotation, where one finds text annotations for images (Barnard et al. 2001). In annotation approaches, a set of keywords are assigned for each image, and information retrieval methods are used to understand the content of images.

Content-based image retrieval uses the visual contents of an image such as color, shape, texture, and spatial layout to represent and index the image. In typical content-based image retrieval systems (Figure 1), multi-dimensional feature vectors are used to describe the visual contents of the images in the database. The most widely used features for color are mean, median, and

standard deviation of red, green, and blue channels of color histograms and for texture features are contrast, energy, correlation, and homogeneity. The feature vectors of the images in the database form a feature database. Then the features of the query example and those in the database are compared and the similarities and differences are calculated and retrieval is performed with the aid of an indexing scheme (Siu and Zhang 2003).

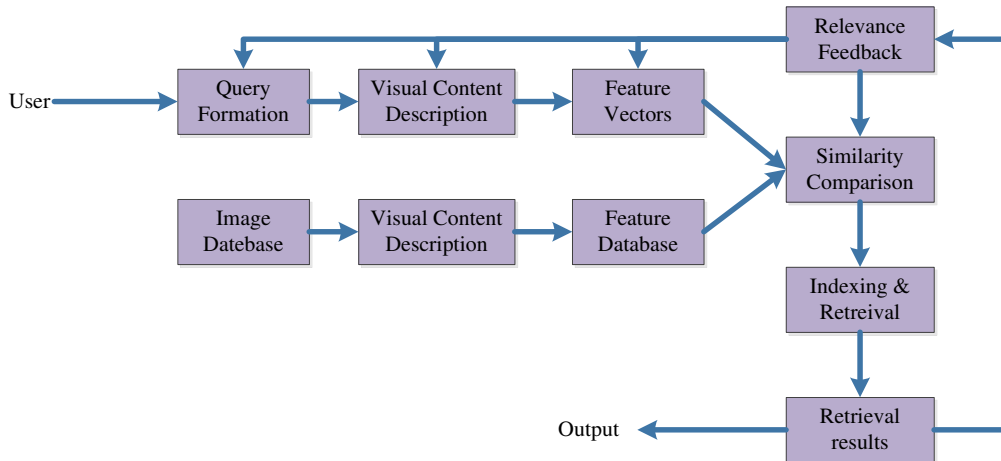


Figure 1: Diagram for content-based image retrieval system (Siu and Zhang 2003)

In CBIR systems, high-level user perceptions cannot be captured by low-level image features (Rui et al. 1998; Sun et al. 2002; Liu et al. 2007); Therefore, region-based retrieval systems (Sumengen et al. 2003) were introduced that attempted to overcome the deficiencies of feature-based image retrieval by representing images at the object-level. A region-based retrieval system applies image segmentation to decompose an image into regions, and if the decomposition is ideal it will correspond to objects (Liu et al. 2004). Region-based retrieval systems segment images into regions and retrieve images based on the similarity between regions. Relevance feedback (Rui et al. 1998; Porkaew et al. 1999; Wu and Zhang 2002) is another approach to reduce the gap between high-level image concepts and low-level image features by involving the

user's perception of images in the retrieval process. In this approach the original query is refined, based on the feedback that the user provides on the retrieval images at each iteration (Shah-hosseini 2007).

### **2.1.1 Review of Existing CBIR Systems**

Many CBIR systems have been built since the early 1990s for annotation of images, illustration of specific text, or set of search and browsing in the image categories. In the following, a few representative systems are listed and their characteristics highlighted.

**Simplicity** (Wang et al. 2001) segments an image into regions and classified the image into semantic categories. To segment an image, it is partitioned into blocks and a feature vector for each block is extracted. The k-means algorithm (MacQueen 1967) is used to cluster the feature vectors into several classes. Each class corresponds to one region in the segmented image. For segmentation six features are used for segmentation. Three features are color components (LUV color space), and the other three represent energy in high frequency bands of the wavelet transform. Each region has significance credit assigned to it which will be used in distance function. The significant factor can be uniform (when all regions are equally important), based on the area percentage, or location of the region.

**Blobworld** (Carson et al. 2002) is another CBIR system that is based on segmenting the image into regions and querying the image database using features of those regions instead of querying on global properties. This system recognizes images as collection of objects that are in a spatial relationship to one another. Using the expectation maximization algorithm to estimate the parameters of this model, the resulting pixel-cluster memberships provide a segmentation of the

image. After the image is segmented, different features such as color and texture are generated for the different segments. In the querying process, the user will be allowed to access the segments directly to determine which features of the image are important to the query. When results are returned, the user also sees the Blobworld representation of the image, which is used to refine the user's query.

**ALIPR**, which stands for Automatic Linguistic Indexing of Pictures, is a machine-assisted image tagging and searching service (Li and Wang 2008). Users can do text searches and provide feedback to the system to find similar images. Users can also upload an image, and the system will perform concept analysis and generate a set of annotations or tags automatically. The system then retrieves images from the database that are visually similar to the uploaded image. In the process of automatic annotation, if the user doesn't think the tags given by the system are suitable, he or she can add other tags to describe the image. This is also the "training" process for the ALIPR system. Annotations are processed using the WordNet to derive a lexical signature for each image. An integrated region based similarity is also calculated between each pair of images. An overall similarity measure is formed using lexical and visual features. At the end, a mutual reinforcement based rank is calculated for each image using the image similarity matrix.

These commercial products and experimental prototype systems are designed to describe color, shape and texture features but cannot adequately represent image semantics and have many limitations when dealing with broad content image databases. According to Eakins there are three levels of queries in CBIR (Eakins 1999):

Level 1: Image retrieval is done by a set of features such as color, texture, shape or the spatial location of elements in the image. An example for this type of query is: "find pictures like this"



Level 2: Image retrieval is based on objects of a given type identified by a set of features, with some degree of logical inference. For example “find a picture with a desk in it”.

Level 3: Image retrieval is based on abstract attributes along with a significant amount of high-level reasoning about the purpose of the objects or scene depicted. This level includes retrieval of images with specific emotion and religion. For example “find pictures of joyful crowd”

Level 2 and 3 are referred to as semantic image retrieval and the gap between levels 1 and 2 is called the semantic gap. Users using the level 1 queries are usually required to submit an example image or sketch as the query. But it is more convenient for users to submit keywords instead of sample images. Therefore, to support query by high-level concepts, a CBIR systems should provide full support in bridging the ‘semantic gap’ between numerical image features and the richness of human semantics.

## **2.2 Multimedia Analysis Using Kinect**

One of the problems of image processing based methods is inverting 2D to 3D. This is difficult since the information for depth has been lost for each point in the image. On the other hand presenting clutters and varying lighting, viewpoint and exposure setting are challenging. The researchers in the image processing area try to improve object recognition by adding features and increasing interactive rate with users.

After the invention of the Microsoft Kinect in 2009, many developers began researching possible applications of Kinect that go beyond the system’s original intended use in playing video games.

Depth camera was not conceptually new. But what Kinect does is helps researchers develop immersive applications that capture voice, movement and gesture recognition. Researchers have become more and more interested in using Kinect because of its low cost and the extensive built-in image processing capabilities of the device (Newcombe et al. 2011).

In Figure 2, internal components of the Kinect are shown. As shown in the figure it has sensors and a light source that are used to capture RGB and depth data. The color camera supports a maximum resolution of  $1280 \times 960$  while the depth camera supports a maximum resolution of  $640 \times 480$  (Webb and Ashley 2012). The microphone array is shown which is composed of four different microphones.

Several free software development libraries are readily available to researchers (CLNUI, OpenNI or Microsoft windows SDK) which enable applications to access and manipulate depth, color and speech data with all the libraries needed for data processing (Weerasinghe et al. 2012). These platforms can be used for creating a broad range of applications.

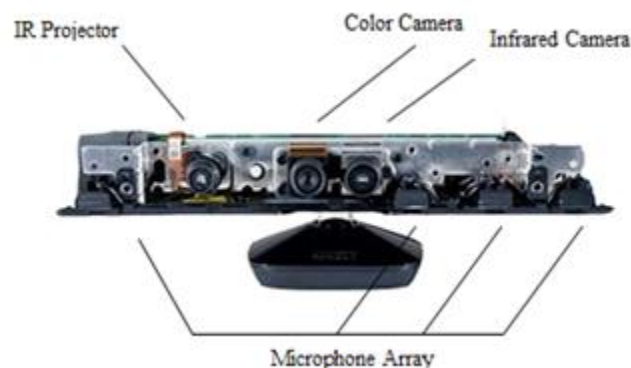


Figure 2: The Kinect components

The RGB camera captures color images and IR camera and the IR projector form a stereo pair with a baseline of approximately 7.5 cm. The IR projector releases a fixed pattern of light and dark speckles. The IR camera reads the IR beams reflected back to the sensor.

Depth is calculated by triangulation against a known pattern from the projector. The pattern is memorized at a known depth. The reflected beam is converted into depth information which measures the distance of the object in world coordinates from the sensor.

One of the areas that need fundamental advances in sensor technology is indoor scenes. In recent years, researchers in computer graphics and image processing have paid a lot of interest to extracting features using active sensors, stereo cameras, range scans and time-to-flight cameras (Henry et al. 2010). Laser scanning is another method for capturing depth and finding the point cloud. This method needs additional infrastructure and is time consuming to set up. Some of the technologies are expensive and cannot be used for all materials and objects. Due to advantages discussed above, Kinect is an emerging trend of technologies that provide high quality features consisting of depth and color. The applications can execute in real-world visual perception (Khoshelham and Elberink 2012).

Researchers have studied different ways of labeling objects in an image by running machine-learning techniques on three dimensional point cloud data retrieved from both real world environments and Google 3D warehouse (which stores 3D models of objects). This way, Google 3D warehouse can recognize objects in the real world based on the shape of the model objects (Lai and Fox 2010). Researchers developed a method for detecting 3D objects in RGBD-images and extracting representations for robotics tasks (Richtsfeld and Vincze 2009; Holz et al. 2011).

In the work done by Du et al. (2011) a Kinect sensor is used to construct a 3D model for an indoor environment on a mobile laptop by computing inter frame alignment. They tried to implement interaction with the users by providing online visual feedbacks to the users.

Similarly, in the work done by Newcombe et al., 3D surface model for indoor was constructed. However, instead of aligning the depth frames, they simultaneously estimated the camera pose and tracked live depth images against the constructing model (Newcombe et al. 2011). They showed the reconstruction results with low drifting and high accuracy and demonstrated the advantage of tracking against a global model over frame-to-frame alignment.

Lai et al. published an object dataset and showed the impact of depth information in object detection. They used histogram of oriented gradients over both color and depth images. They proposed an approach that segments objects from video using depth maps and incorporates temporal information through optical flow to correct the motion artifacts in the depth data (Lai et al. 2011).

Silberman proposed an approach for indoor scene segmentation. The proposed methodology uses a CRF-based model to evaluate the range of different representations for depth information. In their work they concentrated on finding main segmentations of the image and studied the effect of the main segmentation on other segments (Silberman et al. 2012).

Kinect is also being used in the field of construction. RGBD sensors can be used in applications such as surveillance and assessment of construction sites, tracking of materials, equipment and labor, safety and reconstruction (Weerasinghe et al. 2012; Kamat et al. 2010; Tang et al. 2010; Lytle 2011; Golparvar-Fard et al. 2010).

To summarize, while the quality of this depth map is generally remarkable given the cost of the device, a number of challenges still remain. From the emergence of Kinect in 2009, research has been done on utilizing depth data. Many different approaches developed to solve and increase the accuracy of object recognition and segmentation. In some of the studies an existence of a specific object is being assumed. For example in the work done by Weerasinghe detecting hardhat is used as a method for finding labor in a scene (Weerasinghe et al. 2010). Another example for object recognition is combining shape and functionality of objects. Eguchi and Thompson implemented an object recognition system by considering both shape and functionality of an object (i.e. ‘a place to sit’ can be a functionality of chair) (Eguchi and Thompson 2012; Grabner et al. 2011).

It needs to be emphasized that the approach used in this methodology is not limited to a specific set of objects and can be applied in a general setting. The system is also automated in which no interaction with user is needed.

### **2.3 Features and Feature Selection**

Image content may include both visual and semantic content. Visual content includes color, texture, shape, spatial relationship. Semantic content is obtained either by textual annotation or by complex inference procedures based on visual content (Siu and Zhang 2003; Guldogan and Gabbouj 2008).

A visual content descriptor can be either global or local. A global descriptor uses the visual features of the whole image, whereas a local descriptor uses the visual features of regions or objects to describe the image content. Local descriptors have proven to be more useful for object

recognition within cluttered scenes and with partially occluded objects. For obtaining the local visual descriptors, an image is divided into segments.

Descriptors are used for creating searchable indexes of image features. The characteristics of a good descriptor are being unique, rapidly generated, compact, invariant, and efficient to match (Siu and Zhang 2003).

The Scale-Invariant Feature Transform (SIFT) (Lowe 1999) is an algorithm to detect and describe local features in images. SIFT is an approach for detecting and extracting local feature descriptors that are reasonably invariant to changes in illumination, image noise, rotation, scaling, and small changes in viewpoint. In this method a feature detector which identifies key points in an image with an associated region (known as an oriented disk) with assigned scale and orientation. These features are identified by first generating a scale space using Gaussian smoothing and resolution reduction and then differencing the scale images. Key points are generated for extreme regions in a neighborhood. Low contrast and edge based key points are rejected, and then orientations for each point are assigned from the histograms of the gradients in an interest point's neighborhood. SIFT descriptors compactly encode each feature into a 128-element vector, and have been shown to be one of the most effective descriptors.

SIFT descriptors are computed at sparse, scale-invariant key image points and are rotated to align orientation (Lowe 1999). On the other hand HOG computes gradients in the region and put them in bins according to orientation (Dalal and Triggs 2005).

HOG computes the discretized gradients by using a filter to twist the image region. The region is then segmented into a dense grid of uniformly spaced cells. For each cell, a histogram of

gradients is computed. For each pixel a vote is casted which is weighted by the strength of its gradient and distance to the center of the cell, and each vote casts toward a certain gradient orientation range corresponding to a bin in the histogram. Finally, each histogram is contrast normalized over spatial neighbors (Dalal and Triggs 2005).

In the following section, the low level features that are used in CBIR systems are explained.

### **2.3.1 Color**

The color feature is one of the most widely used visual features in image retrieval. In image retrieval, the color histogram is the most commonly used color feature representation. Besides the color histogram, several other color feature representations have been applied in image retrieval, including color moments and color sets (Tasic et al. 2003). A color descriptor metric indicates the similarity, or equivalently, the dissimilarity of the color features of images by measuring the frequency count of the intensities. Any color could be represented by a linear combination of the three primary colors (R, G, and B) or any color spaces. Therefore these three colors are used to describe visible colors by representing these as vectors in 3D RGB color space.

It should be noted that in most of the CBIR works, the color images are not pre-processed. Since color images are often corrupted with noise due to capturing devices or sensors, applying effective filters for removing the noise will significantly improve retrieval accuracy (Plataniotis and Venetsanopoulos 2000; Lukac et al. 2005).

### **2.3.2 Texture**

Texture generally refers to the presence of a spatial pattern that has some properties of homogeneity (Smith and Change 1996). Basically, texture representation methods can be classified into two categories: structural and statistical (Choras 2007). Structural methods which include morphological operator and adjacency graph, describe texture by identifying structural primitives and their placement rules. They tend to be most effective when applied to textures that are very regular. Statistical methods that characterize texture by the statistical distribution of the image intensity include Fourier power spectra, co-occurrence matrices, shift-invariant principal component analysis (SPCA), Tamura feature, Markov random field, fractal model, and multi-resolution filtering techniques such as Gabor and wavelet transform (Choras 2007).

Tamura et al. characterized image texture along the dimensions of coarseness, contrast, directionality, likeliness, regularity, and roughness (Tamura et al. 1978). Among the various texture features, Gabor features and wavelet features are widely used for image retrieval. The wavelet transform is a multi-resolution approach. Wavelet transform refers to the decomposition of a signal with a family of basis function which is obtained through translation and dilation of a special function called the mother wavelet. The computation of 2D wavelet transform involves recursive filtering and sub sampling.

### **2.3.3 Shape**

Shape features include aspect ratio, circularity, Fourier descriptors, moment invariants, consecutive boundary segments (Mehrotra and Gary 1995). Shape features have shown to be useful in many domain specific images such as man-made objects. Shape features are usually



described after images have been segmented into regions or objects. The state of art methods for shape description can be categorized into either boundary-based (rectilinear shapes (Sadeghi and Farhadi 2011), polygonal approximation (Russell et al. 2008) and Fourier-based shape descriptors (Felzenszwal et al. 2010) or region-based methods (Deng and Manjunath 2001).

#### **2.3.4 Depth**

One of the most important tasks for a computer vision system is calculating the distance of various points in the scene relative to the position of the camera. Using techniques of projective geometry from multiple viewpoints are common methods for extracting depth information from intensity images. To use this method, assumptions need to be made about the imaged scene. Typical techniques involve analyzing features in intensity images using parallel lines and vanishing points to determine the affine structure of the scene, however it is not yet possible to have these techniques function fully automatic (Scholz-Reiter et al. 2011; Kosecka and Zhang 2006).

Depth images can be produced using Kinect. The primary function of Kinect is producing three-dimensional data. A value is provided at each pixel which is a function of the distance of the corresponding point in the scene from the sensor.

Active technique and triangulation are two of the most commonly used principles for obtaining such depth. Time-of-Flight (TOF) is an example of an active technique. In order for TOF camera to calculate depth, a light is released to the scene and the reflection of the light is measured with the reference signal (Kolb et al. 2010). To obtain the depth information the measurement is correlated with the modulated light. A pre-generated map of infrared dots is projected and the reflection of this pattern of dots is then received by the COMS camera (Mutto et al. 2012). The

distance of the objects in the scene is then calculated by comparing the received pattern with a pattern sent. The depth sensor from Kinect returns a 11-bit raw data number which is then further processed to get the true depth (Webb and Ashley 2012).

## **2.4 Overview of Machine Learning Approaches**

Machine Learning (ML) is essential for automatic systems to make decisions and to infer new knowledge about the world. In Table 1 some of the most important methodologies currently used in the field of machine learning are listed. The main machine learning techniques are divided into supervised learning (such as Support Vector Machines) and unsupervised learning (such as K-means clustering).

### **2.4.1 Supervised Methods**

Supervised learning such as support vector machine (SVM) and Bayesian classifier are often used to learn high-level concepts from low-level image features. Support Vector Machine is a binary classification method based on statistical learning theory which maximizes the margin that separates samples from two classes (Burges 1998; Cortes and Vapnik 1995). This supervised learning machine provides the option of evaluating the data under different spaces and functions through Kernels that range from simple linear to Radial Basis Functions (Chang and Lin 2001; Burges 1998; Cortes and Vapnik 1995). Additionally, its wide use in the field of machine learning research and ability to handle large feature spaces makes it an attractive tool for object recognition, text classification, etc., and is considered a good candidate for using in image retrieval system (Chang and Lin 2001; Tong and Chang 2001). Another widely used learning method is Bayesian classification (Vasconcelos and Lippman 1997). Using binary Bayesian

classifier, high-level concepts of natural scenes are captured from low-level image features (Vailaya et al. 2001; Crandall and Felzenszwalb 2005).

Table 1: Machine learning techniques (Calix 2011)

Technique	Definition	Pros	Cons
Support Vector Machines	Supervised learning approach that optimizes the margin that separates data.	SLT Confidence characteristic (expected risk)	class imbalance issues
Decision Trees	This method performs classification by constructing trees where branches are separated by decision points.	Easy to understand	Not flexible
Neural Networks	Model represents the structure of the human brain with neurons and links to the neurons.	Versatile	Can obscure the underlying structure of the model
K-means clustering	Unsupervised method that forms k-means clusters to minimize distance between centroids and members of cluster.	Unsupervised – so no training needed	Needs clearly defined separations in the data in order to be effective
Linear Discriminant Analysis (LDA)	Creates linear function of features to classify data	Simple yet robust classification method	Normality assumptions of the classes
Gaussian Mixture models (GMM)	This probabilistic method represents signals as weighted sums of normal distributions	Can be used to represent non-normal distributions	Initialization is important for optimization
Naïve Bayes	Probabilistic Learning: to calculate the probability of seeing a certain condition in the world selecting the most probable class given the feature vector	Fast, easy to understand the model	Bayes assumptions of independence
Maximum Likelihood Estimation (MLE):	Calculating the likelihood that an object will be seen based on its proportion in the sample data	Simple	Too simplistic for some applications
Expectation-Maximization	Similar to MLE but is used when there is missing data in the training set	Very useful when missing data	
Hidden Markov Models (HMM)	A Markov Chain is a weighted automaton consisting of nodes and arcs where the nodes represent states and the arcs represent the probability of going from one state to another.	Probabilistic	Combinatorial complexity
Bayesian Networks	Probabilistic networks	Graphical representation improves understanding	Requires knowledge of probabilities

### **2.4.2 Unsupervised Methods**

Unlike supervised learning in which the presence of the outcome variable guides the learning process, unsupervised learning has no measurements of outcome, the task is rather to find out how the input feature are organized or clustered (Liu et al. 2007). Image clustering is the typical unsupervised learning technique for retrieval purposes. It groups a set of image data in a way to maximize the similarity within clusters and minimize the similarity between different clusters. Each resulting cluster is associated with a class label, and images in same cluster are supposed to be similar to each other.

Cluster-based retrieval of images by unsupervised learning (CLUE), is a new approach developed in which the system aims to retrieve images by including the similarity knowledge between target images through user interaction (Chen et al. 2005). They claim that the degree of user involvement with CBIR systems can help reduce the semantic gap.

The K-means algorithm (MacQueen 1967) is one of the most popular methods in clustering based on optimization quality of clusters. In this method, the center vector of each cluster (in mass cluster representation) is employed to minimize the sum of internal-cluster distances. Li and Wang developed a new algorithm based on statistical modeling and optimization (D2-clustering), in which data points (region-based image signatures) are characterized by a set of probability weighted vectors. D2-clustering aims to generalize k-means algorithm by using sets of weighted vectors instead of vectors. Their method for real-time automatic image annotation attempts to establish probabilistic relationships between images and relevant labels (Li and Wang 2008).

## 2.5 Image Segmentation

In image processing, finding an object in the entire image is computationally expensive. The problem is detecting objects in image without knowing their identities in advance. The goal of a segmentation algorithm is to identify parts of an image that are likely to correspond to individual objects. More precisely, image segmentation is the process of assigning a label to every pixel in an image in such a way that pixels with the same label share certain visual characteristics.

In a recent work, regions are used for object detection instead of the traditional sliding window approach (Gu et al. 2009). A strong assumption was made that each segment represents a (probabilistically) recognizable object part. Gould et al. provided a complete description of the scene using dynamically evolving decompositions that explain every pixel (both semantically and geometrically) (Gould et al. 2009). However, the method cannot distinguish between foreground objects and often leaves them segmented into multiple dissimilar pieces. Liu et al. used a non-parametric approach to image labeling by warping a given image onto a large set of labeled images and then combining the results (Liu et al. 2009).

Some systems design their own segmentations in order to obtain the desired region features during segmentation, be it color, texture, or both (Wang et al. 2001; Mezaris et al. 2003; Town and Sinclair 2000). These algorithms are usually based on k-means clustering of pixel/block features. In Wang et al's work, first an image is segmented into small blocks of size  $4 \times 4$  from which color and texture features are extracted. Then k-mean clustering is applied to cluster the feature vectors into several classes with each class corresponding to one region. Blocks in same classes are classified into same regions.

**Blobworld segmentation** (Carson et al. 2002) is another widely used segmentation algorithm (Shi et al. 2004; Liu et al. 2007). It is obtained by clustering pixels in a joint color-texture-position feature space. First, the joint distribution of color, texture, and position features is modeled with a mixture of Gaussians. Then expectation maximization (EM) algorithm is used to estimate the parameters of the model. The resulting pixel-cluster membership provides a segmentation of the image. The resulted regions correspond roughly to objects.

Discriminating between textures is the main difficulty in a segmentation method (Deng and Manjunath 2001; Liu et al. 2009). Many texture segmentation algorithms require the estimation of texture model parameters which is a very difficult task (Deng and Manjunath 2001). ‘**JSEG**’ **segmentation** (Deng and Manjunath 2001) overcomes this problem. Instead of trying to estimate a specific model for texture region, it tests for the homogeneity of a given color-texture pattern. ‘JSEG’ consists of two steps. In the first step, image colors are quantized to several classes. By replacing the image pixels by their corresponding color class labels, a class-map of the image is obtained. Spatial segmentation is then performed on this class-map which can be viewed as a special type of texture composition. The algorithm produces homogeneous color-texture regions and is used in many systems (Feng and Chua 2003; Liu et al. 2005).

**Contour models or snakes** known as active contour model is one of the popular techniques for image segmentation (Chen et al. 2010). A snake is an energy-minimizing spline guided by external constraint forces and is influenced by image forces that pull it toward features such as lines and edges. Snakes are active contour models; they look onto nearby edges and localize them accurately (Kass et al. 1988).

**Region growing** is a segmentation algorithm that starts with seeded pixels, and then adds neighboring pixels to those seeded region sets that are within a feature metric distance (e.g. within an intensity-level tolerance) (Deng and Manjunath 2001). There are unseeded versions of the algorithm. Region growing algorithms are highly dependent upon the similarity metric used, which is in turn dependent upon the application.

**Connected components** is a segmentation algorithm that is similar to region growing but employs a graph-based technique for segmentation. The basic premise of connected components is to form regions that contain pixels of the same value. Best results are obtained with a binary image. Connected components will yield labeled segments that are both the same in intensity and space whereas an algorithm such as k-means yields those similar in intensity. The algorithm starts by first considering each pixel in the binary image as a node in a graph. The objective is to find groups of neighboring pixels that have the same intensity. This first starts by taking each foreground pixel (intensity of 1) and looking at either the four or eight immediate neighbors to determine if they are the same value. Each neighbor with the same value is given a region label. This process repeats until there are no more neighbors with the same value.

**Graph partitioning** examines each pixel as a node and the distance to neighboring pixels as edges (Shi and Malik 2000). Edges that are more than a distance metric will be removed and the image is then segmented. This approach is particularly useful for segmenting the 2.5D range images.

Graph-based image segmentation techniques generally represent the problem in terms of a graph  $G = (V; E)$  where each node  $v_i \in V$  corresponds to a pixel in the image, and the edges in  $E$  connect certain pairs of neighboring pixels (Felzenszwalb and Huttenlocher 2004). Each edge

will have a weight which is based on properties of the pixels that it connects, such as their image intensities. Depending on the method, there may or may not be an edge connecting each pair of vertices. The earliest graph-based methods used fixed thresholds and local measures in computing segmentation. In the work done by Zahn a segmentation method is presented which is based on the minimum spanning tree (MST) of the graph (Zahn 1971). This method has been applied both to point clustering and to image segmentation. For image segmentation the edge weights in the graph are based on the differences between pixel intensities, but for point clustering the weights are based on distances between points (Felzenszwalb and Huttenlocher 2004).

Depending on the dataset used and the requirements needed for the system, different segmentation algorithms can be used and therefore it is hard to judge which algorithm is the best. For the purposes of this research modification are made to the graph-based segmentation by adding parameters that are discussed in Chapter 4.

## 2.6 Surface Detection

For finding surface, RANSAC (RANDOM SAMPLE CONSENSUS) procedure is used (Fischler and Bolles 1981). The plane containing the points  $px^1, \dots, px^N$  can be modeled by linear equations:

$$\begin{cases} \theta_1 \times px_1X + \theta_2 \times px_1Y + \theta_3 \times px_1Z + \theta_4 = 0 \\ \vdots \\ \theta_1 \times px_NX + \theta_2 \times px_NY + \theta_3 \times px_NZ + \theta_4 = 0 \end{cases} \quad (\text{Eq. 1})$$

We can write these equations in a matrix form:



$$\begin{bmatrix} \text{px}_1^T & 1 \\ \vdots & \vdots \\ \text{px}_N^T & 1 \end{bmatrix} \theta = A\theta = 0 \quad (\text{Eq. 2})$$

Therefore the parameters of the plane that contain the points  $\text{px}_1, \dots, \text{px}_N$ , are given by solving the following cost function.

$$\theta^* = \underset{\substack{\theta \in \mathbb{R}^4 \\ \|\theta\|=1}}{\text{argmin}} \|A\theta\|^2 \quad (\text{Eq. 3})$$

This cost function can be solved using SVD decomposition of A. for finding the best match and comparing different hypothesis planes on points, error is calculated. The estimated error is defined as the squared distance between the point  $\text{px}$  and its orthogonal projection onto the plane. For each point  $\text{px}$  the unique solution of  $\underset{\text{px}' \in \text{plane}}{\text{argmin}} \|\text{px} - \text{px}'\|^2$  can be obtained using the method of Lagrange multipliers (Hartley and Zisserman 2000).

## 2.7 Scene Detection

While image understanding is still very much an open problem, much progress is currently being made in scene classification. Because scenes can often be classified without full knowledge of every object in the image, the goal is not as ambitious as object recognition. For instance, if a person recognizes chair under the table, he may hypothesize that he is looking at a classroom scene, even if he cannot see every detail in the image. On the other hand, if he sees many sharp vertical and horizontal edges, he may be looking at an urban scene. It may be possible in some cases to use low-level information, such as color or texture, to accurately classify the scene. In other cases, object recognition may be necessary, but not necessarily of every object in the scene. In general, classification seems to be an easier problem than unconstrained image understanding.

By using the local low-level feature detectors across large regions of the visual field, global feature inputs are estimated and the scene can be classified based on the feature vector (Oliva and Torralba 2006). Mulhem presented a novel variation of fuzzy conceptual graphs for use in scene classification (Mulhem et al. 2001). A fuzzy conceptual graph is composed of three semantics. Steeves et al. have shown that an individual with a profound visual form agnosia (i.e. incapable of recognizing objects based on their shape) could still identify scene pictures from colors and texture information only (Steeves et al. 2004). In the modeling presented by Oliva and Torralba, they only consider global features of receptive fields measuring orientations and spatial frequencies of image components that have a spatial resolution between 1 and 8 cycles per image (Oliva and Torralba 2006). Lipston's approach which is called configural recognition uses relative spatial and color relationships between pixels in low resolution images to match the image with class models (Lipston et al. 1997). Some systems also use 'pseudo-object-based' features. They use segmented images and calculate features from each region, but do not explicitly perform object recognition. The Blobworld system (Carson et al. 2002), developed at Berkeley, was created primarily for content-based indexing and retrieval, but is also used for scene classification.

## **2.8 Object Recognition**

In this part we focus on the methods which are suitable representation of the original data (approximating the original data by keeping as much information as possible). Therefore, objects can be described by different cues. These include model-based approaches, shape-based approaches, and appearance-based models (Roth and Winter 2008). Model-based approaches try to represent the object as a collection of three dimensional, geometrical primitives (boxes,

spheres, cones, cylinders, generalized cylinders, surface of revolution) whereas shape-based methods represent an object by its shape/contour. In contrast, for appearance-based models only the appearance is used, which is usually captured by different two-dimensional views of the object-of-interest. Based on the applied features these methods can be sub-divided into two main classes: local and global approaches. A local feature is a property of an object located on a single point or small region. But global features try to cover the information content of the whole image (e.g. mean values or histograms of features).

Most of the local appearance based object recognition systems work on distinguished regions in the image. Currently most popular distinguished region detectors can be divided into three categories (Roth and Winter 2008).

1. Corner based detectors: locate the point of interests and regions which contain a lot of image structure (e.g., edges), but they are not suited for uniform regions and regions with smooth transitions.
2. Region based detectors: regard local blobs of uniform brightness as the most salient aspects of an image and are therefore more suited for the latter.
3. Other approaches: for example take into account the entropy of a region (Entropy Based Salient Regions).

In the following the most popular appearance based object recognition algorithms are listed (Roth and Winter 2008):

- Harris or Hessian point based detectors (Harris and Stephens 1988; Mikolajczyk and Schmid 2001).
- Difference of Gaussian Points (DoG) detector (Lowe 2004).

- Harris or Hessian affine invariant region detectors (Harris-Affine) (Mikolajczyk and Schmid 2002).
- Maximally Stable External Regions (MSER) (Matas et al. 2004).
- Entropy Based Salient Region detector (EBSR) (Kadir et al. 2003; Kadir et al. 2004).
- Intensity Based Regions and Edge Based Regions (IBR, EBR) (Tuytelaars and Van Gool 2004).

There have been other part-based recognition methods, which like the pictorial structure approaches are based on separately modeling the appearance of individual parts and the geometric relations between them. However most of these part-based methods make binary decisions about potential part locations (Felzenszwalb and Huttenlocher 2005). Most part-based methods use some kind of search heuristics, such as first matching a particular “distinctive” part and then searching for other parts given that initial match, in order to avoid the combinatorial explosion of the configuration space. Such heuristics make it difficult to handle occlusion, particularly for those parts that are considered first in the search. Being able to use part recognition needs a precise segmentation and since in the methodology used in this research the segmentation is not perfect then parts cannot be recognized. For this reason a new approach is proposed and explained in Chapter 5 for object recognition without being effected by the errors in segmentation.

Traditionally, researchers have divided occlusion reasoning into subtasks of segmentation and line labeling to be solved separately (Hoiem et al. 2011). Modern segmentation algorithms attempt to partition the image according to color or texture similarity or gestalt cues, but the resulting boundaries often do not correspond to complete objects. Figure/ground labeling

algorithms work well, but only when given perfect segmentations (Hoiem et al. 2011). Handling occlusion is a part of object recognition that cannot be separated. Using surface index and contextual cueing help handle this issue (explained in Section 4.3).

Gao et al. presented a Bayesian inference algorithm for image layer representation with mixed Markov random field (Gao et al. 2007). The key contribution of this work is that the 2.1D sketch was formulated using mixed random fields and presented an inference algorithm to solve region coloring/layering and assignments of open bonds simultaneously. Hoiem et al. proposed an algorithm for finding major occlusion boundaries in an image and assigning figure/ground labels to them (Hoiem et al. 2011). In this model, the strategy is to begin with a conservative over-segmentation which consists of thousands of regions and then slowly removes boundaries based on predictions from learned models. As the regions grow, spatial support for computing features improves, and certain features become much more useful. This model got an accuracy of 48.6% on images from LableMe dataset. Desai handled occlusion by changing the weight of object detectors. In the case of occlusion, the CRF learns a positive weight that reinforces both detections related to each occluded objects (Desai et al. 2009). Felzenszwalb et al. introduced an efficient algorithm for finding the best global match of a large class of pictorial structure models to an image which can handle good amounts of noise and occlusion (Felzenszwalb et al. 2005).

To summarize, there have been a variety of techniques proposed and implemented for object recognition. Although the results of the state-of-art is not even close to human detections, but much progress has been done in the area of feature descriptors and classifying them and incorporating semantically meaningful information into the detection pipeline. Our approach provides a new set of features and descriptors for object recognition.

## 2.9 Commonsense Knowledge

Knowledge corpora refer to databases that include information about words or concepts. There are several implementations such as: ConceptNet (Havasi 2007), WordNet (Miller et al. 1990) , FrameNet (Baker et al. 2002) and YAGO (Suchanek et al. 2007).

ConceptNet gathers commonsense knowledge thorough ordinary people in its site. The data is represented in the form of semantic network and is available for use in natural language processing. It also has a Python implementation which gives access to a copy of ConceptNet database. The version of ConceptNet 3.0, for instance, contains over one million assertions collected by human annotators from the World Wide Web. And as it is shown in Table 2 the semantic relations are embedded in different categories.

ConceptNet represents the information as a directed graph. The nodes of the graph are the concepts and the labeled edges are assertions of common sense that connect two concepts. Each assertion is associated with a frequency value that defines whether people said that the relationship is sometimes, generally or always true. The frequency value can also be negative defining that the relationship is rarely or never true. Since the information is gathered from humans, the system needs to handle noise and incorrect information and imprecision. Therefore AnalogySpace process was developed that represented the knowledge in a semantic network as a sparse matrix. The concepts are along one axis and the features along another axis. By using singular value decomposition (SVD) the dimensionality is reduced and the result represents the most salient aspects of the knowledge (Speer et al. 2008).

Table 2: ConceptNet semantic relations

Category	Semantic relations
Things	Is A Property Of Has Property Part Of Made Of Has A
Events	First Sub Event Of, Last Sub Event Of Has Prerequisite Event For Goal Event Event For Goal State Event Requires Object Has Sub Event
Actions	Effect Of Effect Of Is State Capable Of Receives Action Causes
Spatial	Often Near Location Of At Location
Goals	Desires Event Desires Not Event Motivated By Goal
Functions	Used For
Generic	Can Do Conceptually Related To

The format of output produced by ConceptNet for a word such as “cat” is shown in Figure 3.

AtLocation (cat, lap) []
AtLocation (cat, bed) []
AtLocation (cat, windowsill) []
CapableOf (cat, hunt mouse) []
CapableOf (cat, eat mouse) []
CapableOf (cat, drink water) []
CapableOf (cat, corner mouse) []
HasA (cat, four leg) []
HasA (cat, whisker) []
HasA (cat, fur) []
IsA (cat, carnivore) []

Figure 3: ConceptNet selected output for "cat"

## 2.10 Corpora

A database with images and corresponding labels is needed to represent each image with a set of keywords (scene, objects). Most of the researchers use PASCAL images to be able to compare

their image annotations with the state of the art models in PASCAL dataset (Everingham et al. 2006; Van and Zisserman 2010). There are available algorithms which allow researchers to label images and share the annotations with the rest of the community. The dataset from these algorithms can be used to train the model for each object. The most common used algorithms for sharing image datasets are LabelMe (Russell et al. 2008), ImageNet (Deng et al. 2009), Lotus Hills (Yao et al. 2007), flickr and Corel image datasets (Li and Wang 2003). LabelMe provides an online annotation tool to build a large database of annotated images. In flickr the images include rich descriptions such as titles, tags, locations and more to give your photo context and a life of its own. ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds of other images. Lotus Hills and Corel's image database contain large number of images in various contents.

Unfortunately there is no dataset with RGBD information of indoor sites and the ones that exist are video files and can't be used for this research.



## **CHAPTER 3 - AUTOMATED SEMANTIC CONTENT EXTRACTION APPROACH**

This research develops a computationally efficient framework for identifying objects in commercial building environments. The following sub problems were solved in developing our model (Figure 4):

- Development of a new segmentation method by generating features for each pixel. The feature vector is a combination of graph-based segmentation, surface index and feature vector for each pixel.
- Object recognition using supervised classification methods with a new set of features extracted from each segment.
- Developing a scene detection model based on global features using color, texture and local histogram of gradient.
- Developing an evidence fusion model for combining the multiple learning models .This model takes into consideration the information from scene, objects and commonsense knowledge information.

We utilize a bottom-up approach, gathering feature information for each pixel in the image and then determining the related pixels (Figure 5).

A hybrid approach is developed that combines local features that capture the relationship between each pixel and its neighbor pixels as well as global features.

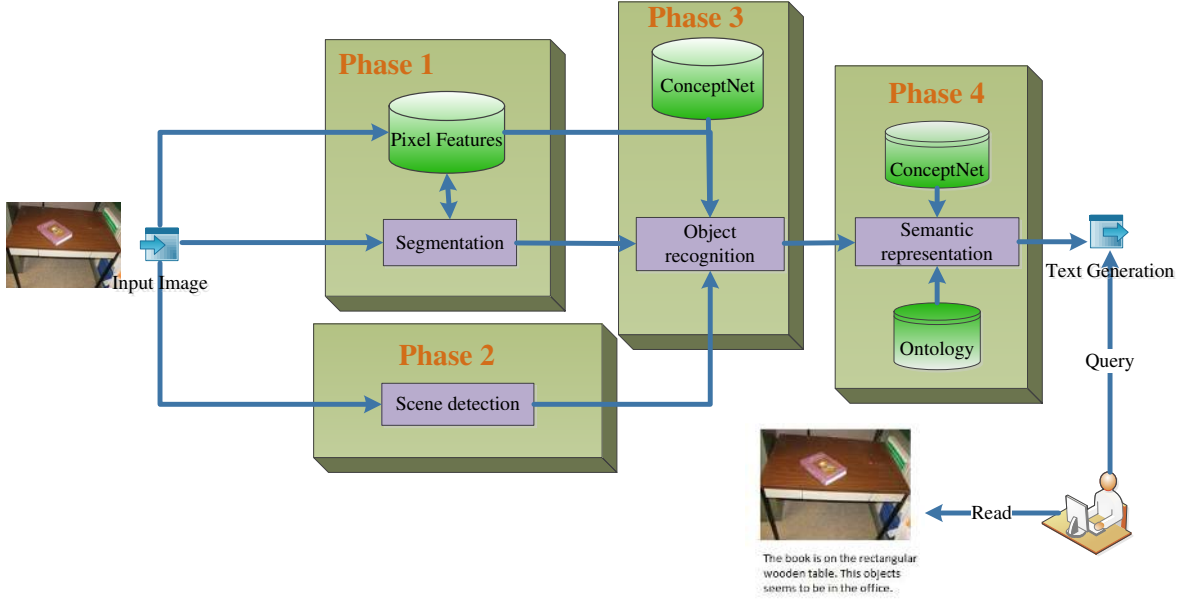


Figure 4: Diagram of the framework

In this dissertation the goal is fusing information to take advantage of both types of methodologies to improve accuracy and efficiency. An image consists of objects and structures. Segmentation is a fundamental process of this methodology and any error in this part reflects on semantic image processing. Therefore we developed an approach to capture and label perceptually important regions which often reflect global representation and understanding of the image.

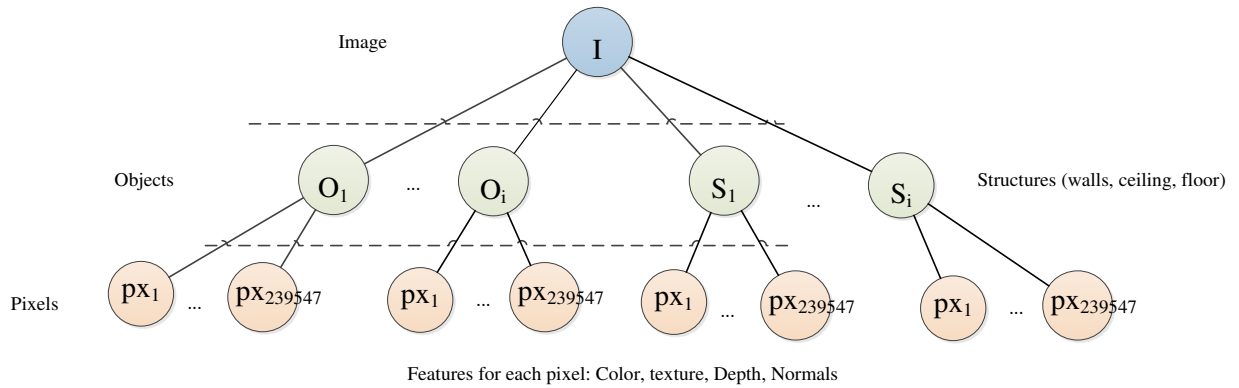


Figure 5: Image hierarchy

In Chapter 4, a new methodology is proposed for segmenting the image into potential objects. The main structures of the image such as the walls and floor are identified in this section. The new segmentation methodology developed in this research modifies Felzenszwalb's cost function (Felzenszwalb and Huttenlocher 2004) and combines the result with surface index and depth, color, texture and normal features to overcome issues of occlusion and shadow commonly found in images. In Chapter 5, new features such as height, width, length are added to color, texture and histogram of gradient and used in the classification method for object recognition. In Chapter 6, local gradients and global color and texture features are generated to identify the scene and capture important non-local properties of an image. In Chapter 7, we implemented the use of contextual and common sense information in improving the object recognition and scene detection, and fused information of scene and objects to reduce the level of uncertainty.

### **3.1 Tools**

The following tools are used for performing the tasks required in this methodology: Matlab; Calibration Toolbox ([http://www.vision.caltech.edu/bouguetj/calib\\_doc](http://www.vision.caltech.edu/bouguetj/calib_doc)); Meshlab; Python 2.6; Microsoft Visual Studio 10; RGBDemo Toolbox (<http://labs.manctl.com/rgbdemo/index.php>); ConceptNet (Havasi et al, 2007); libfreenect ([http://openkinect.org/wiki/Main\\_Page](http://openkinect.org/wiki/Main_Page)); and WEKA (Witten and Frank 2005).

### **3.2 Corpus**

In this dissertation image understudying approach is implemented on images from indoor and outdoor construction sites. To our knowledge there is no corpus that contains these types of images and for this reason we created our own dataset for scene detection. For scene detection, a

dataset consisting of 764 images of classrooms, bathrooms, computer labs, corridors and outdoors were collected. Each of these images was annotated and a scene tag was assigned to each image. The annotations were done manually by two annotators and since the scenes were discriminable by humans, there was 100% agreement between annotators.

For object recognition, there is also no image dataset with RGBD information of indoor sites. Based on our knowledge only video files are available which have RGBD data, which are not suitable for this research since the data are biased and redundant. For this reason it was necessary to manually construct training and testing corpus.

For the object recognition task, a dataset of 200 single object images were captured. The dataset consists of images of chairs, tables, couches, trash bins, computers, sinks, toilets and cars. The images were captured using Kinect sensor and Photoshop was used to remove the background so all the images consist of one object on a white background. To improve the classification, an “unknown” class was also created using random objects with different color, scale, orientation and shape. An example of the object data set is given in Table 3.

Another dataset was developed for testing the methodology, and consists of multi object images in different scenes. The images were taken manually using Kinect sensor from buildings on the LSU campus. This dataset contains 40 images which are taken from classrooms, computer labs, bathrooms, corridors and outdoors. OpenKinect ([http://openkinect.org/wiki/Getting\\_Started](http://openkinect.org/wiki/Getting_Started)) is used for capturing images using Kinect. There are many open source libraries available that will enable the Kinect to be used. Libfreenect is the core library for accessing the Microsoft Kinect camera. Depth and RGB data is extracted for each image and saved in individual files with ppm

and pgm extensions. The size of RGB data is  $480 \times 640 \times 3$  and the size of depth data is  $480 \times 640$ .

Table 3: Example of object dataset consisting single object images in  $427 \times 561$  pixels of size on a white background



## CHAPTER 4- SEGMENTATION

In this chapter a new set of features are extracted that are used in clustering methods. The new segmentation methodology developed in this research modifies Felzenszwalb's cost function (Felzenszwalb and Huttenlocher 2004) and combines the result with surface index and depth, color, texture and normal features to overcome issues of occlusion and shadow commonly found in images.

After clustering, each image is divided into potential object regions and structures. The results of this section are then used in the object recognition process explained in Chapter 5.

An overview of the system is given in the following.

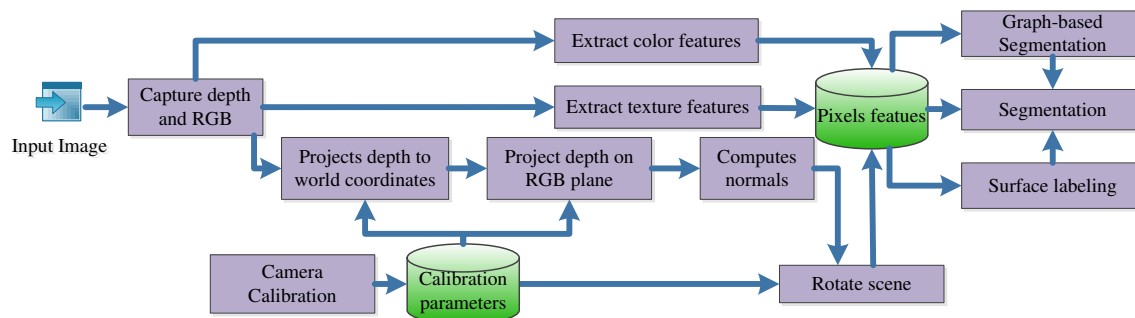


Figure 6: Segmentation engine

The feature vector used for segmentation consists of three major groups of features, which are extracted for each pixel:

1. Pixel feature vector consisting of color, XYZ in world coordinates, normal, and texture (see Section 4.2).
2. Surface index (explained in Section 4.3).
3. Graph based region index (explained in Section 4.4).

Before extracting the feature vectors, preprocessing is performed. One issue in using the Kinect is that there is an offset between the different cameras and therefore it needs preprocessing to map depth data to RGB data. In the following section, the process used for calibrating the Kinect camera is discussed.

#### 4.1 Camera Calibration

In order to generate 3D points from Kinect's color and depth camera, the system must first be calibrated. This includes internal calibration for each camera as well as relative pose calibration between the two cameras. The Kinect device uses an infrared camera to detect a projected dot pattern. However, it returns a processed image that is not aligned with the original infrared image. The best match gives an offset from the known depth, in terms of pixels which is called disparity. For each pixel, the distance to the sensor can be retrieved by considering the corresponding disparity. Our setup consists of a depth and color camera and both of them are calibrated simultaneously.

From the intercept (Thales') theorem we have:

$$\frac{D}{b} = \frac{z_o - z_k}{z_o}, \quad \frac{d}{f} = \frac{D}{z_k} \quad (\text{Eq. 4})$$

where  $z$  is the depth (in meters),  $D$  is the horizontal baseline between the cameras (in meters),  $f$  is the (common) focal length of the cameras (in pixels),  $b$  is the base length and  $d$  is the disparity (in pixels).

In addition, integration of the color image and IR image is needed due to differences between camera characteristics. For this purpose images from a chessboard should be captured using both

the IR and RGB cameras. For this calibration a chessboard with pattern size<sup>2</sup> of 0.25 centimeters is printed out and glued on a board (as shown in Figure 7).

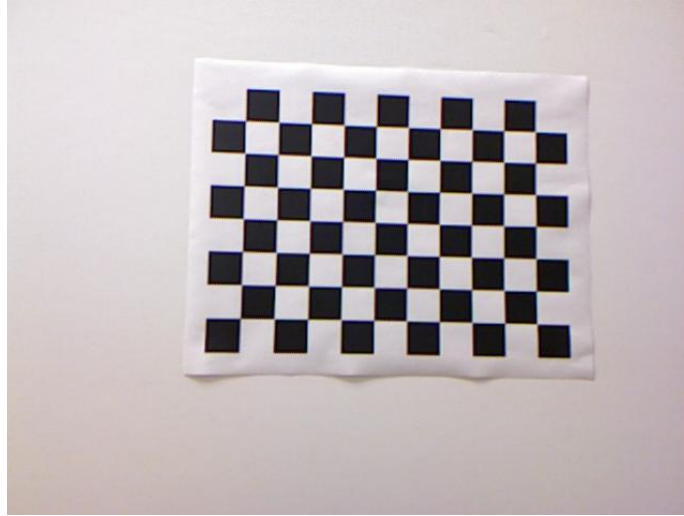


Figure 7: Calibrate the Kinect using chessboard

The RGBDemo toolbox is used to first grab images from the chessboard and then simultaneously calibrates two color cameras, a depth camera, and the relative pose between them.

In order to ensure an optimal calibration the following points have to be taken into consideration when taking images:

- Image areas, especially the image corners must be covered in the images taken.
- For better precision, the chessboard should be as close to the camera as possible.
- For depth calibration, images with both the IR and depth are needed. The calibration algorithm will automatically determine which grabbed images can be used for depth calibration.
- Different images of the chessboard with various angles should be taken.

---

<sup>2</sup> The pattern size corresponds to the size in meters of one chessboard square.



- An average of 30 images is adequate for calibration.
- To ensure the calibration has succeeded the re-projection error should be less than 1 pixel. If the error is significantly higher, it means the calibration has probably failed.

The checkerboard corners are extracted from the color image. A homography is then computed for each image using the known corner positions in world coordinates and the measured positions in the image. Each homography then imposes constraints on the camera parameters which are solved with a linear system of equations. The distortion coefficients are initially set to zero. The same method is used to initialize the depth camera parameters.

Figure 8 shows the images with the detected corners.

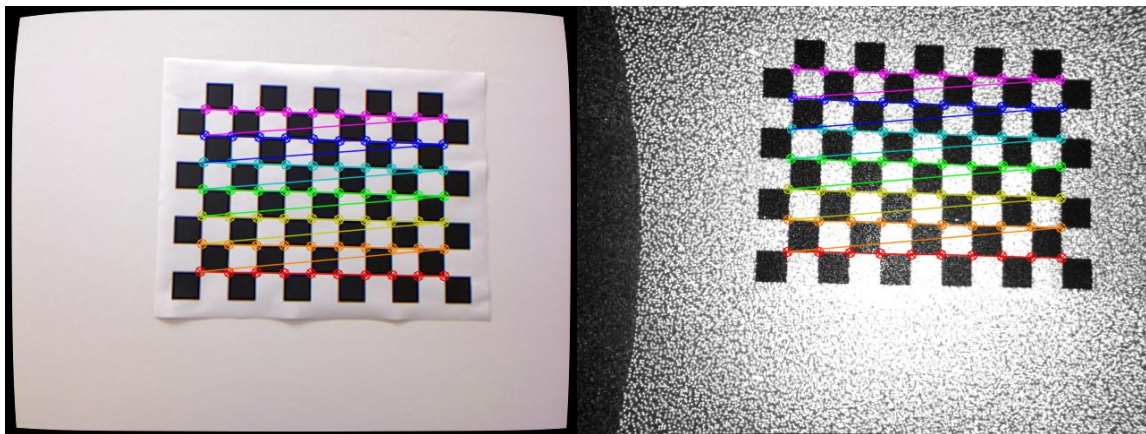


Figure 8: Left-Calibration of color camera; Right-Calibration of infrared camera

Using Matlab Calibration toolbox, the intrinsic camera matrices, camera calibration matrix, and distortion parameters of the two cameras are obtained. The geometrical relationship between IR and color cameras is computed by retrieving extrinsic parameters from the calibration toolbox.

The IR camera and the IR projector form a stereo pair with a baseline of approximately 7.5 cm. The IR and RGB cameras are separated by a small baseline. The external transform between the two cameras need to be calibrated.

Typical translation values are  $[0.0254 \quad -0.00013 \quad -0.00218]$ . The measured distance between IR and RGB lens centers is about 2.5 cm horizontally.

In the three devices we tested, the rotation component of the transform was also very small. Typical offsets were about 0.5 degrees, which translates to a 1 cm offset at 1.5 m. The internal parameters are:

- **Focal length:** The focal length in pixels is stored in the  $(f_x, f_y)$ .
- **Principal point:** The principal point coordinates are stored in the  $(c_x, c_y)$ .
- **Distortions:** The image distortion coefficients (radial and tangential distortions) are stored in the  $\{ k_1, k_2, p_1, p_2, k_3 \}$ .

The calibration parameters that were estimated for the Kinect used here are based on the work done by Nicolas Burrus (2012) and the calibration results after optimization are listed in Table 4. The calibration parameters generated here are later used in fitting depth data to color data.

## 4.2 Pixel Features

In this section, the process of generating a feature vector for each pixel is explained. The feature vectors are then used in a classification algorithm to specify the regions of potential objects.

The calibration parameters are used for projecting depth on RGB plane and finding XYZ in the world coordinates. Normal vector for each pixel is calculated, and by finding the vanishing points of each image the image can be rotated such that the structure of the room has the highest normals (explained in detail in Section 4.2.6). For each pixel 3 features for color, 4 features for

texture, 3 features for normals and 3 features for location of each pixel in the world coordinate are extracted.

Table 4: Calibration parameters for depth and color cameras

Calibration matrix for color camera	Calibration matrix for depth camera
fx_rgb: 5.2921508098293293e+02	fx_d: 5.9421434211923247e+02
fy_rgb: 5.2556393630057437e+02	fy_d: 5.9104053696870778e+02
cx_rgb: 3.2894272028759258e+02	cx_d: 3.3930780975300314e+02
cy_rgb: 2.6748068171871557e+02	cy_d: 2.4273913761751615e+02
k1_rgb: 2.6451622333009589e-01	k1_d: -2.6386489753128833e-01
k2_rgb: -8.3990749424620825e-01	k2_d: 9.9966832163729757e-01
p1_rgb: -1.9922302173693159e-03	p1_d: -7.6275862143610667e-04
p2_rgb: 1.4371995932897616e-03	p2_d: 5.0350940090814270e-03
k3_rgb: 9.1192465078713847e-01	k3_d: -1.3053628089976321e+00
<b>Rotation matrix between two cameras:</b>	
$\begin{bmatrix} 9.9984628826577793e-01 & 1.2635359098409581e-03 & -1.7487233004436643e-02 \\ 1.4779096108364480e-03 & 9.9992385683542895e-01 & -1.2251380107679535e-02 \\ 1.7470421412464927e-02 & 1.2275341476520762e-02 & 9.9977202419716948e-01 \end{bmatrix}$	
<b>Translation matrix between two cameras:</b>	
$[1.9985242312092553e-02 \quad -7.4423738761617583e-04 \quad -1.0916736334336222e-02]$	

#### 4.2.1 Capturing Depth and RGB

Kinect application development starts with the Kinect sensors. Data retrieval from sensors is from a set of events on the Kinect sensor object. First the streams are enabled and after determining that all frames are available at once, the application starts the Kinect sensors by calling start method.

The Kinect sensor captures color and depth frames simultaneously. The sensors that help capture depth data consist of an infrared laser emitter and an infrared camera which provides a  $480 \times 640$  depth map in real time. For capturing color images RGB camera is used which supports different

image formats including RGB and YUV with different FPS (frame per second) and resolutions. In this research, the RGB format and  $480 \times 640$  resolution is used.

Raw depth values are integer values between 0 and 2047. They can be transformed into depth in meters using Equation 5. The parameters are valid for all of the Kinect sensors and are given on the ROS( Robot Operating System) Kinect page ([http://www.ros.org/wiki/kinect\\_node](http://www.ros.org/wiki/kinect_node)).

$$\text{Raw depth in meters} = 1.0 / (\text{raw depth value} * -0.0030711016 + 3.3309495161) \quad (\text{Eq. 5})$$

A problem in using the Kinect is alignment of depth and color data. In addition, the depth map is only valid for objects that are more than 2.5 feet and less than 13 feet away from the sensing device. If an object is closer than 2 feet to the Kinect or farther than 13 feet from the Kinect its depth will not be detected correctly and will be presented as a zero depth (Webb and Ashley 2012). As with the color camera, occlusion can also occur with the depth camera. Due to the offset between the color and depth cameras, when merged together, "shadowing" may occur along the profiles of objects, where the object has blocked a portion of the infrared from reaching surfaces behind it and thus the shadow has zero depth values.

Figure 9 shows an example image where the blue points are pixels which Kinect was not able to detect depth information. Although the lack of depth information for these points causes a decrease in the performance of the segmentation process, other features can still be used to resolve the segmentation. Color and texture information and graph based region index are available for each pixel. Surface index and normals are calculated based on a patch of neighborhood points and missing depth information for part of the points in the patch, wouldn't affect the features' accuracy.

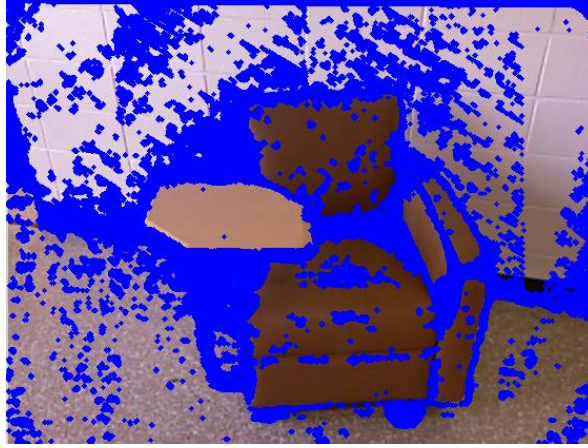


Figure 9: Showing points with no depth data

Another limitation of Kinect is estimating depth. The captured 3d points from the depth sensor have error. The scanner has a maximum error of 10 cm at 4 m away, and less than 2 cm off within 2.5 m (Rafibakhsh et al. 2012).

#### 4.2.2 Projecting Depth on RGB Plane

The RGB camera and depth camera are not physically located in the same spot, and therefore result in stereo vision problems. This makes it very challenging to determine the color of a depth pixel.  $p(x,y)$  in color image is not the same  $p(x,y)$  in the depth image. For integrating the depth and intensity, the intrinsic data parameters for both cameras and the extrinsic mapping between the cameras are needed. The calibration matrixes that were generated in the calibration process (Section 4.1) are used here to map depth and color information for pixels together.

The first step is to undistort color and depth images using the estimated distortion coefficients. Then, using the depth camera intrinsic parameters from the calibration process, each pixel  $(x_d, y_d)$  of the depth camera can be projected to metric 3D space using Equation 6 (Burrus 2012):

(Eq. 6)

$$\begin{aligned}P3D.x &= (x_d - cx_d) * \text{depth}(x_d, y_d) / fx_d \\P3D.y &= (y_d - cy_d) * \text{depth}(x_d, y_d) / fy_d \\P3D.z &= \text{depth}(x_d, y_d)\end{aligned}$$

where  $fx_d$ ,  $fy_d$ ,  $cx_d$  and  $cy_d$  are the intrinsics of the depth camera. We can then re-project each 3D point on the color image and get its color using Equation 7.

(Eq. 7)

$$\begin{aligned}P3D' &= R.P3D + T \\P2D_{\text{rgb}.x} &= (P3D'.x * fx_{\text{rgb}} / P3D'.z) + cx_{\text{rgb}} \\P2D_{\text{rgb}.y} &= (P3D'.y * fy_{\text{rgb}} / P3D'.z) + cy_{\text{rgb}}\end{aligned}$$

R and T are the rotation and translation parameters estimated in Section 4.1 during the stereo calibration. Because of the calibration it is guaranteed that depth and color information for each point are mapped together (Figure 10).

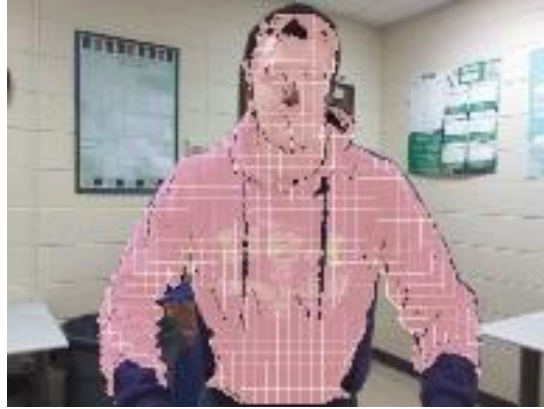


Figure 10: Projecting depth on color image

#### 4.2.3 Finding XYZ in World Coordinates

Once the distance is calculated using the measurements in the previous section, Equation 8 is used to get a good approximation for converting (i, j, z) to (x,y,z).

(Eq. 8)

$$x = (i - w / 2) * (z - 10) * 0.0021 * (w/h)$$

$$y = (j - h / 2) * (z - 10) * 0.0021$$

$$z = z$$

The parameters are valid for all of the Kinect sensors and are given on the OpenKinect page ([http://openkinect.org/wiki/Imaging\\_Information](http://openkinect.org/wiki/Imaging_Information)). Figure 11 shows the 3D reconstruction of points in space for the picture in the left.

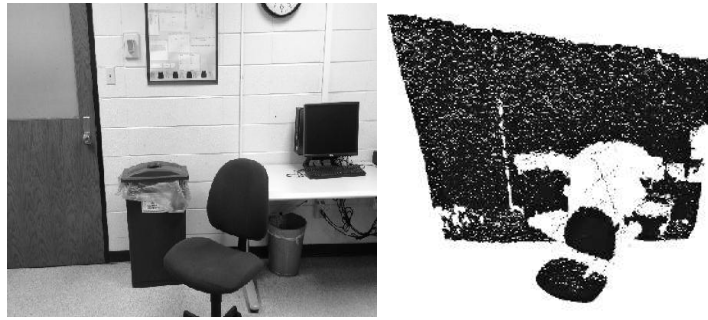


Figure 11: Showing XYZ in world coordinates of chair in front of wall

#### 4.2.4 Computing Normals

Pixel normal is important properties of a geometric surface and shows the vector perpendicular to the surface in that point. The problem in determining the normal for each pixel in image is approximated by the problem of estimating the normal of a plane tangent to the points, which in turn becomes a least-square plane fitting estimation problem. This problem can be solved by analyzing the eigenvectors and eigenvalues of a covariance matrix created from the nearest neighbors of the query point. After finding first three eigenvalues, they first need to be normalized and then the geometric information of the surface is produced.

Matrix of surface normal is generated for each point (p) in 3D point cloud by getting the nearest neighbors of p and computing the surface normal (n) of each point.

To extract a surface normal at each pixel, planes on point clouds were fitted. The equation of the plane is given by:

$$n_x \times px^iX + n_y \times px^iY + n_z \times px^iZ + n_D = 0 \quad (\text{Eq. 9})$$

where  $n = [n_x \ n_y \ n_z]$  is normal orientation for a given point cloud.

We solved ‘n’ for each pixel point based on eigenvectors and eigenvalues of a covariance matrix created from the neighborhood of the pixel point (explained in detail in Section 2.6). This process is equivalent to finding the axis system in which the covariance matrix is diagonal. The eigenvector with the largest eigenvalue is the direction of greatest variation, the one with the second largest eigenvalue is the (orthogonal) direction with the next highest variation and so on.

#### 4.2.5 Finding Vanishing Points

For determining the relative orientation of the camera with respect to the scene, the vanishing points should be determined (Gallagher 2005). Based on the images used in here, the majority of lines are aligned with the three principal orthogonal directions of the world coordinate system. To obtain candidates for the principal directions we used Rother’s algorithm (Rother 2002). First image derivatives are computed followed by the non-maximum suppression using the Canny edge detector (Canny 1986) (Figure 12). The gradient direction of each edge is quantized. This algorithm finds orthogonal vanishing points with robust voting and search schemes based on line segments. Rother ranks all triplets  $(vp_1, vp_2, vp_3)$  using a voting strategy, which scores angular deviation between the line and the point. Candidate points are chosen as intersection points of all detected lines. This step is referred to as the accumulation step.



In the next step, the dominant clusters of line segments are searched for. The algorithm terminates when dominant vanishing points that fulfill orthogonal criterion, camera criterion and vanishing line criterion are found (Rother 2002).



Figure 12: Detected line segments

#### 4.2.6 Rotating the Scene and Normalizing Features

Once the vanishing points have been detected, the relative orientation of the camera with respect to the world references can be computed. Since the vanishing directions are projections of the vectors associated with the three orthogonal directions  $i, j, k$ , they depend on the rotation matrix and each pixel point on image frame can be transformed to world reference frame. In particular we can write that (Kosecka and Zhang 2006):

$$K^{-1}v_i = Roe_i; K^{-1}v_j = Roe_j; K^{-1}v_k = Roe_k \quad (\text{Eq. 10})$$

where  $K$  is the calibration matrix,  $v_i$  and  $v_j$  and  $v_k$  are vanishing points,  $Ro$  is the rotation matrix,  $e_i$  and  $e_j$  and  $e_k$  are unit vectors associated with the world coordinate frame. Each vanishing direction is proportional to the column of the rotation matrix  $Ro = [r_1; r_2; r_3]$ . By choosing the

two best vanishing directions and normalizing them, the third row can be obtained as  $r_3 = r_1 \times r_2$  by enforcing the orthogonality constraints. Rotational geometrical relationship between the depth and color cameras and rotational mapping from the world reference frame into the camera reference frame are shown in Figure 13.

#### 4.2.7 Texture Features

For extracting texture features, first the RGB image is converted into an HSV color image. In the HSV color space, value refers to the brightness and perceived light intensity. The proposed system extracts grey level co-occurrence matrix from V component (Srinivasan and Shobha 2008).

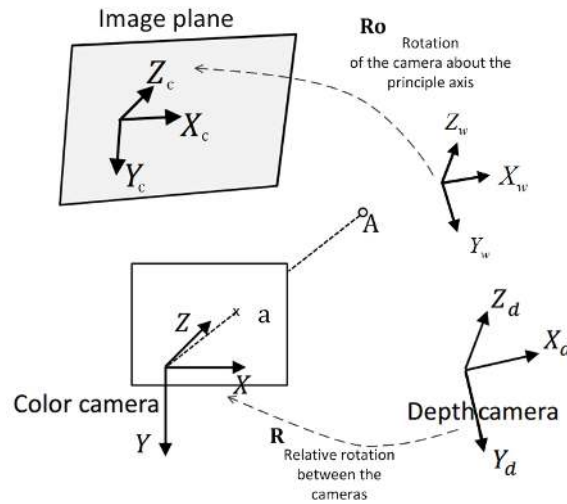


Figure 13: Rotation of the camera and relative rotation between the cameras

The gray-level co-occurrence matrix (GLCM) is one of the most widely used statistical texture measures and considers the spatial relationship of pixels. The idea of the method is extracting the intensity relationships between all pairs of two neighboring pixels. The Gray-level co-occurrence

matrix (GLCM) calculates how often a pixel with the intensity (gray-level) value of ‘m’ occurs in a specific spatial relationship to a pixel with the value ‘n’.

In the following discussion, the default neighborhood relationship is defined as the pixel of interest and the pixel to its immediate right (horizontally adjacent). Each element (m, n) in the resultant GLCM is simply the sum of the number of times that the pixel with value ‘m’ occurs in the specified spatial relationship to a pixel with value ‘n’. Four statistical features are computed on GLCM of V component of the each pixel.

**Contrast** is a measure of intensity contrast between a pixel and its neighbor over the whole image.

$$\sum_{i,j} |i - j|^2 p(i, j) \quad (\text{Eq. 11})$$

**Correlation** is a measure of how correlated a pixel is to its neighbor over the whole image.

$$\sum_{i,j} \frac{(i - \mu_i)(j - \mu_j) p(i, j)}{\sigma_i \sigma_j} \quad (\text{Eq. 12})$$

**Energy** can be calculated by sum of squared elements in the GLCM.

$$\sum_{i,j} p(i, j)^2 \quad (\text{Eq. 13})$$

**Homogeneity** is a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal (MATLAB Image Processing toolbox).

$$\sum_{i,j} \frac{p(i, j)}{1 + |i - j|} \quad (\text{Eq. 14})$$

The contrast, correlation, energy and homogeneity features are calculated and added to the overall feature vector.

### 4.3 Surface

In the previous section, features including the XYZ coordinates and normal for each pixel were generated. In this section, the goal is to identify the main surfaces in the image using the RANSAC algorithm (RANdom SAmple Consensus) (Fischler and Bolles 1981). A surface index is assigned to each pixel in the image that will later be used as a feature in the segmentation process.

RANSAC is used to identify points that belong to a plane with specific parameters in the space. A point is called “inlier” if the distance from the point to a plane is smaller than a threshold. The threshold is the perpendicular distance of points to the plane and can be changed depending on the complexity of the image.

The level of complexity has correlation with the number of inliers. For example man-made environment images such as building are characterized by many flat surfaces. On the other hand images with complex and differential geometry are identified with the curvature of more complex objects.

To be able to recognize surfaces in these complex images the threshold should be reduced. The dataset used here consists of indoor and outdoor images of buildings and consists of many flat

surfaces. We assumed that the complexity of the images in our dataset is constant and therefore the minimum number of inliers is 1000 points.

First a small set of random sample points are selected. For each of these sample points a group of neighbors are selected in horizontal and vertical directions and potential planes are generated using a RANSAC procedure for each sample point. At this point many planes are generated. The algorithm finds the inlier points for each plane.

After a few iterations the algorithm terminates and the final plane parameters are calculated. If the number of inliers for a candidate plane is greater than a minimum number of inliers, the planes are selected as final surfaces and assigned an index value.

As it can be seen in Figure 14, not all points get an index value assigned to them. The reason can be either the lack of depth data or point belonging to surfaces with less than the threshold inliers.

It should be taken into consideration that the normals of each pixel should be close to the normal of the plane parameters. Figure 14 shows the surfaces selected using RANSAC.

As shown in the two bottom images in Figure 15, the surface indexing can handle occlusion problems. Although the object is occluded and there is no neighboring similarity, the system is able to put them in the same surface. Errors can arise, however, when the system clusters unrelated surfaces together. This error is handled by using other features (color and texture) (top two images in Figure 15 show errors).

Another issue is that the system only considers flat surfaces and therefore objects with curved surfaces are not handled. Although not considered here this issue can be handled by solving non-linear equation of surfaces with curvatures.

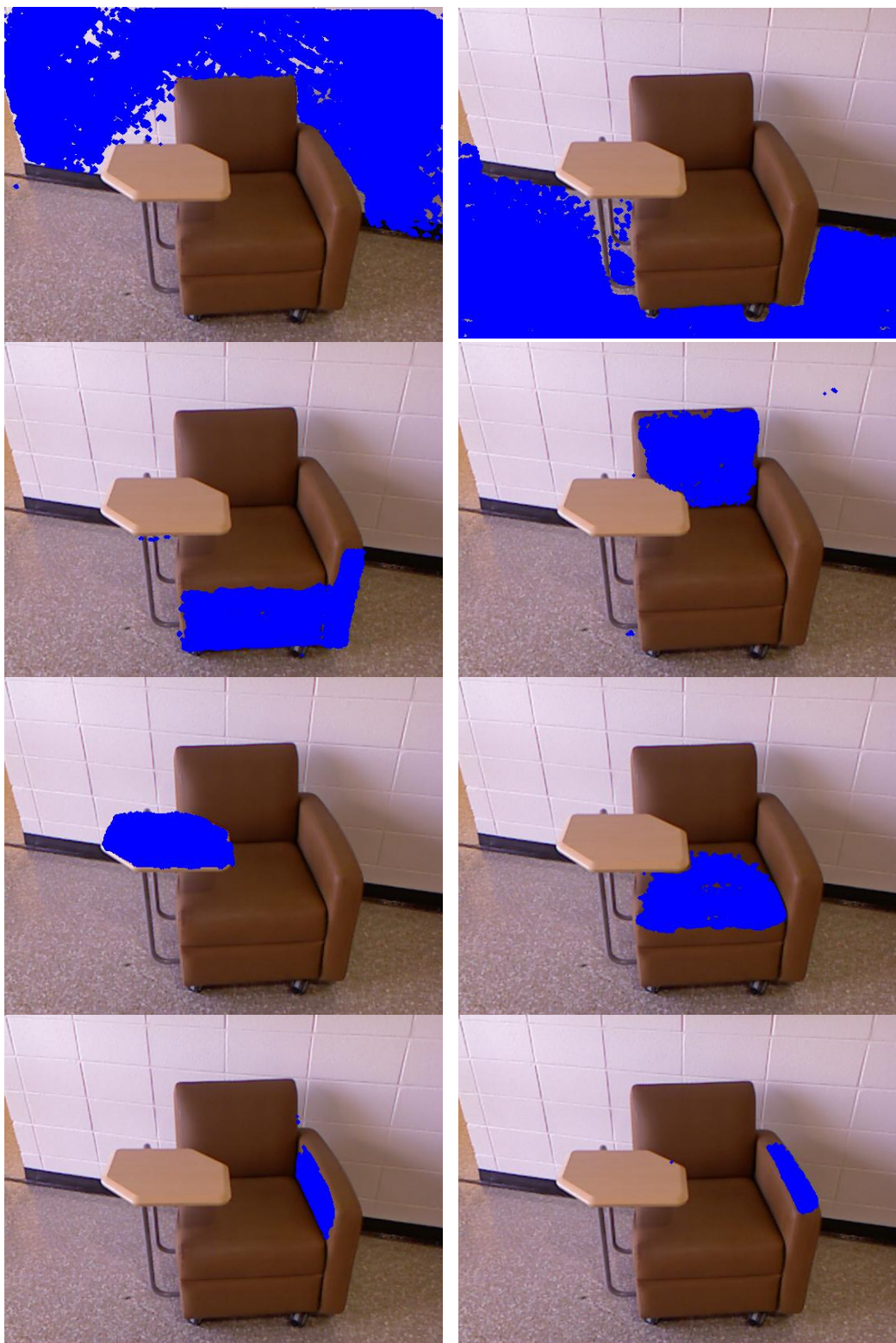


Figure 14: Surface detection for a sample image



Figure 15: Surface detection for a sample image

At this stage a surface index is generated for each pixel point that corresponds to the surface that the point is embedded in. For points for which no surface was found, a random unique surface index is assigned. The reason for using this strategy is to avoid these points getting lumped together and forming one surface.

For specifying room structure (wall, floor, ceiling), surface index plays an important role. The process of identifying structure using surface index is explained in Section 5.1. Surface index is a nominal feature which is used in segmentation and object recognition stages.



#### 4.4 Graph Based Segmentation

An existing graph-based segmentation algorithm of Felzenszwalb and Huttenlocher is used due to higher accuracy and being compatible with our model. Graph-based methods utilize a top-down approach, which models the images as weighted graphs and provides segmentation by recursively partitioning the graph into sub-graphs. In addition, Felzenszwalb and Huttenlocher defined logic for measuring the evidence for a boundary between two regions. The algorithm, written in C++, is open source and available from <http://www.cs.brown.edu/~pff/segment>. Input to the algorithm is individual images in PPM (Portable PixMap) format. Output is also in PPM format, with region pixels color-labeled.

The original algorithm only uses color features to segment the image. Therefore to improve the performance, the original algorithm has been modified by adding depth information in the energy function. Graph based segmentation is based on mapping each pixel to a point in feature space and finding clusters of similar points. The graph  $G(V,E)$  has vertices corresponding to each pixel in feature space and edges  $(v_i, v_j)$  connecting pairs of neighboring feature points  $v_i$  and  $v_j$ . Each edge has a weight  $w(v_i, v_j)$  that measures the dissimilarity between pixel point  $v_i$  and  $v_j$ . Edge weight can be a function of difference between intensity, distance, texture, motion or any local attributes of two pixel points. In segmentation process the points in a region are similar and the points in different regions are dissimilar. This means that the edges between two pixel points in the same region have low weights and edges between two pixel points in different regions have higher weights. Felzenszwalb's edge weight is based on location of the pixel in the image and the color value of the pixel (Felzenszwalb and Huttenlocher 2004). We modified their work by adding depth and normals. We used L2 (Euclidean) distance between points. Normals are



another important attribute for each pixel and reveal information on the orientation of each point. Direction of the normal at each point is the vector perpendicular to the surface in that point. This information can help segment regions with similar geometrid surface. The updated cost function is presented in Equation 15.

$$w(v_i, v_j) = \omega_1 (|I(p_i) - I(p_j)|) + \omega_2 (\sqrt{(p_{ix} - p_{jx})^2 + (p_{iy} - p_{jy})^2 + (p_{iz} - p_{jz})^2}) + \omega_3 (|Nx(p_i) - Nx(p_j)| + |Ny(p_i) - Ny(p_j)| + |Nz(p_i) - Nz(p_j)|) \quad (\text{Eq. 15})$$

The output of the original algorithm is also modified to generate a labeled segment index for each pixel. Figure 16 shows the graph based segmentation for a sample image.

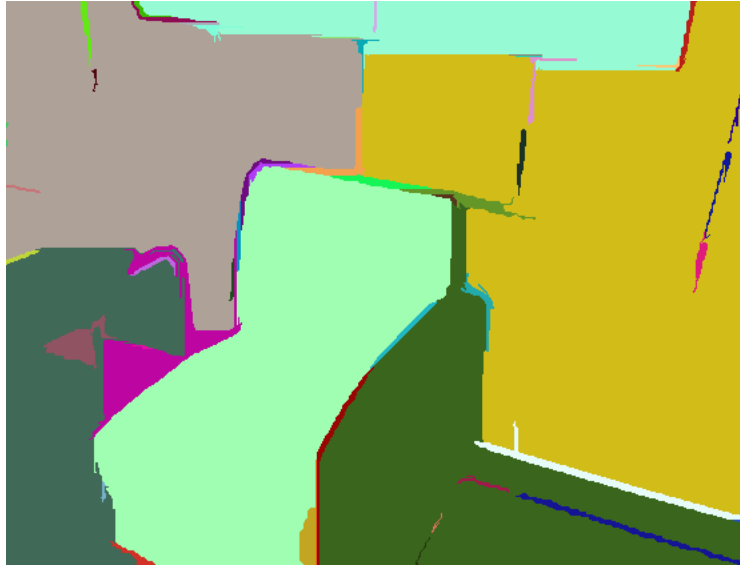


Figure 16: Graph-based segmentation

At this stage a graph based region index is generated for each pixel point. This value is used as a nominal feature for segmentation.

In the following section, the graph based region index is used along with surface index and pixel features in a clustering method to segment the image into regions of potential objects.

## 4.5 Clustering

WEKA (Waikato Environment for Knowledge Analysis) (Bouckaert et al. 2010) is used for applying k-mean clustering method. Using clustering algorithm the image is segmented into regions that correspond to potential objects. To perform clustering, feature vectors were extracted for each pixel in the image. The feature vector consists of the following 15 features:

- 3 color features.
- XYZ world coordinates.
- 4 texture features.
- 3 normal features.
- Surface ID.
- Graph based segment ID.

The data set used here consists of 40 multi object images taken from different scenes. Unsupervised classification is used to cluster the similar data together to form potential object regions that are later used in the object recognition stage (explained in Chapter 5). K-mean clustering with a maximum of 500 iterations and 5 clusters has been used.

The result of this classification is a cluster ID for each pixel in the image. The pixels in the same cluster form regions that are considered to be a potential object. Examples of the images after segmentation are shown in the following Figures.



Figure 17: Segmentation result of an bathroom image

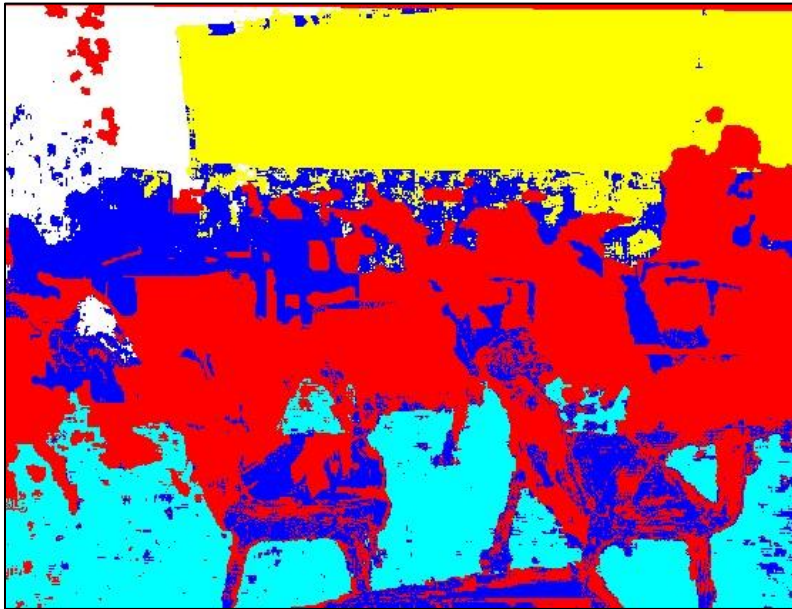


Figure 18: Segmentation of a classroom image

## 4.6 Post-processing of Object Segmentation

After the classification process, the image is segmented into different regions. Before using this information in the object recognition stage, additional post-processing is performed to remove noise from the segments, so as to ensure better performance.

Median filtering (Arias-Castro and Donoho 2009) is used to merge small clusters with the bigger clusters that they are part of. This is done because the goal is finding objects in the image and therefore very small regions shouldn't be considered. These smaller clusters can also be the errors caused by undefined depth points in Kinect.

Since the result of the segmentation is used for object recognition, and in object recognition gradients on each pixel are calculated, it is important to preserve the edges. For this reason, no filtering is used to smooth the edges of the segments. Median filter (Arias-Castro and Donoho 2009) is the only filter used because it is more effective than convolution when we try to simultaneously reduce noise and keep sharpness of edges.

Figure 19 shows slivers in an image. These tiny polygons are eliminated using median filter.

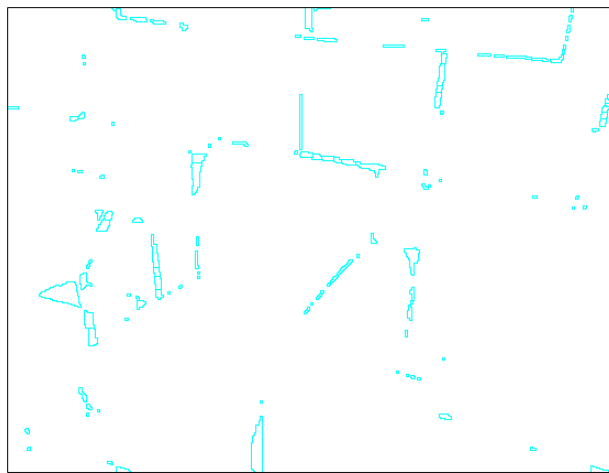


Figure 19: Tiny polygons created in segment boundaries

In Figure 20 an example image is shown before and after applying median filtering.

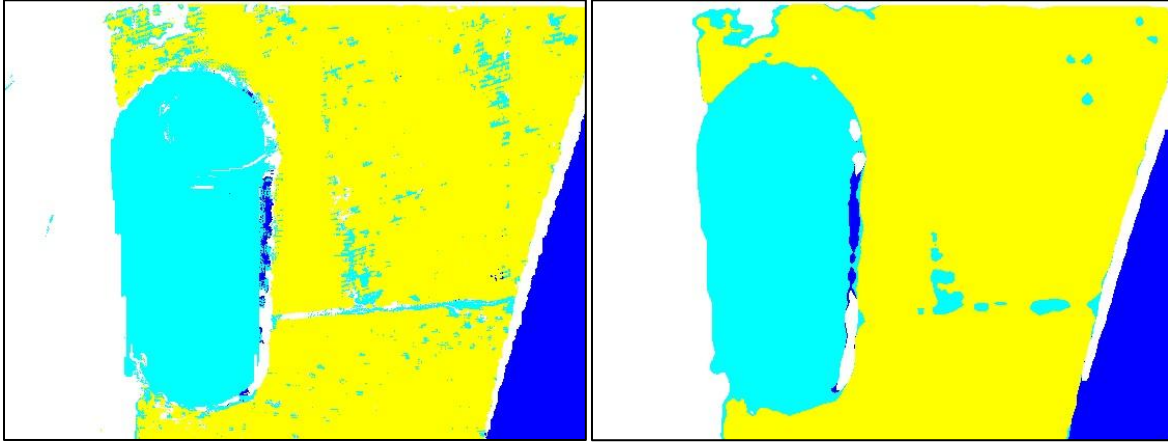


Figure 20: Segmentation result after applying median filtering (right image)

#### 4.7 Time Analysis

The total processing time is 92.6 seconds for each image. The running time of this study can be categorized into generating feature vector for each pixel and clustering. The main time is taken for finding vanishing points and the rotation matrix 3D direction of each line is computed using SVD to find the direction of maximum variance. This process is time consuming to find the highest score principle directions. But the time can be improved significantly since the process is performed on a personal computer with 4 GB Ram and Intel dual core processor. The program is also run using scripts and therefore it's more time consuming. The time breakdown for different stages of the process is shown in Table 5.

Table 5: Time Breakdown

Stage	Time
Extracting pixel features (color, texture, normal, surface ID and XYZ)	84.9
Extracting graph based segment ID	4.3
Clustering	3.4
Total	<b>92.6</b>

## CHAPTER 5- OBJECT RECOGNITION

In this chapter first the structures in an image are specified. These structures can be wall, floor, ceiling and partitions. Surface index ID, normal and rotation matrix are used to find the surfaces. After applying rotation matrix to each of the point normals, the vanishing line will be parallel to the horizontal lines of the image. Based on the normals, surfaces can be labeled as the structure of the image. By finding the structures in an image the 3D structure of the image can be reconstructed. Also relationship between structure and objects (e.g. clock on the wall) can be found and used to obtain 3D localization and labeling of objects.

At this point the image has been segmented to regions that correspond to potential objects. Each region is extracted from the image and is embedded in a  $427 \times 561$  image with white background. Each region is fit into a box and its height, width and length are calculated. These new features are combined with color, texture and HOG features and used in object recognition process.

### 5.1 Finding Room Surfaces

In the previous chapter a surface index was generated for each pixel (Section 4.3). The relative orientation of the camera with respect to the scene was computed using vanishing points (Section 4.2.5). Therefore each pixel point can be transformed to XYZ world coordinates.

This way the points that are inliers on the floor or ceiling have a  $N_z$  (z-normal) close to 1. For the inliers on the wall, depending on the direction of the wall, either  $N_x$  or  $N_y$  has a value close to 1. According to the results from the image it was shown that the surfaces which one of their normals is close to 1, and the minimum number of inliers is greater than 1000 point, were

structures (walls/partitions, floor, ceiling) in the image. Therefore these points were extracted from the image and object recognition process was not applied to them.

In the following Figure, examples are given of the images in which its structures were extracted. As it can be seen in Figure 21 some point on wall and floor surfaces are not specified. The reason is the lack of depth data for these points.



Figure 21: A classroom with its floor and wall specified

In Figure 22 the system was able to identify floor and one of the walls. The reason for not identifying the other walls is that these walls are too close to the camera and therefore depth information cannot be calculated by Kinect.



Figure 22: A bathroom with specified structures

## 5.2 Generating Feature Vector for Each Object

For recognizing objects in an image, a new set of features were generated which are invariant on different scale and orientation. We took advantage of having depth and tried to find an approximation of dimensions for each region which corresponds to a potential object.

Each potential object is embedded in a cube and the length, width and height of the cube are calculated and used as features. Length, width and height represent the geometry based attributes of each object. By averaging texture features for each pixel and finding the standard deviation for contrast, correlation, homogeneity and energy, 8 texture features are extracted for each potential object. The next 36 components represent the color histogram features, which were described in Chapter 2.

For capturing gradient structure, histogram of gradient representation (HOG) is used (Figure 23). HOG computes gradients in the region and put them in bins according to orientation.



HOG computes the discretized gradients by using 1D centered point discrete derivative mask in both the horizontal and vertical directions (Dalal and Triggs 2005) (Equation 16).

$$A_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}, A_y = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}^T \quad (\text{Eq. 16})$$

The region is then segmented into 8 by 8 cells. For each cell, a histogram of gradients is computed. For each pixel a vote is casted which is weighted by the gradient magnitude and orientation. Each vote is cast toward a certain gradient orientation range corresponding to a bin in the histogram. Number of bins are 36 for each cell. Finally, each histogram is contrast normalized over spatial neighbors.

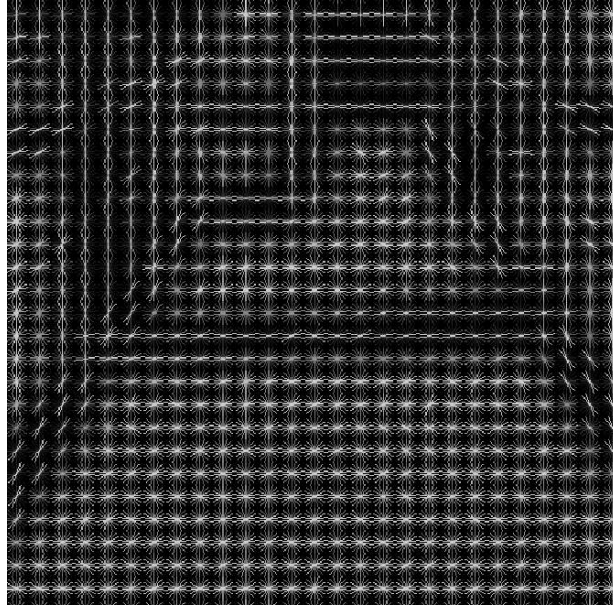


Figure 23: Histogram of gradient

In Table 6 the features used for object recognition are listed. The elapsed time for generating the feature vector for each object is 92.541290 seconds.

Table 6: Feature vector for object recognition

Name	Features	Number of Features
ft(1)	height	1
ft(2)	width	1
ft(3)	length	1
ft(4)	t_con_mean	1
ft(5)	t_con_std	1
ft(6)	t_cor_mean	1
ft(7)	t_cor_std	1
ft(8)	t_e_mean	1
ft(9)	t_e_std	1
ft(10)	t_H_mean	1
ft(11)	t_H_std	1
ft(12:47)	COLOR	36
ft(48:14951)	HOG	18*23*36=14904

### 5.3 Object Feature Analysis

A new set of features are extracted and used for object recognition. In this section the importance of these features are studied by first testing the features on a dataset consisting single object images and then analyzing the features using Chi-square feature ranking technique.

WEKA (Waikato Environment for Knowledge Analysis) (Bouckaert et al. 2010) is used for applying classification methods such as Naïve Bayes, Support Vector Machines and Decision Trees. WEKA is a machine learning software with a collection of machine learning algorithms for data mining tasks.

The object dataset is used for the testing and training of the object recognition system. This dataset consists of 200 images of single object images on a white background. Feature vector consisting of features explained in Section 5.2 is generated for each image. A class is also specified for each image.

For the purpose of evaluation the corpus is divided into two sections: one section for training purposes and the other section for testing purposes. The training set consists of 80% of the corpus and testing consists of the remaining 20%. 10 Fold cross validation is performed to increase the accuracy of the result. For classification tasks, metrics such as precision ( $\text{true positive}/(\text{true positive} + \text{false positive})$ ), recall ( $\text{true positive}/(\text{true positive} + \text{false negative})$ ) and F-measure which is a combination of recall and precision are calculated and used for evaluating the predicted results.

Different supervised classification methods were applied and Bagging using a fast decision tree learner classifier performed the best.

This classifier builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning (with backfitting). The classification results are presented in Table 7 and show an overall F-measure of 84.4% which shows promising results compared to state of art methodologies. Based on the PASCAL challenge which the goal is to recognize objects in 20 object classes in realistic scenes, the accuracy is between 20 to 30 %. The corpora that used in this research has 9 object classes and background is removed in preprocessing.

Among the classes, chair and computer had the highest performance with 97.6% and 95.2% accuracy, and sink had the lowest accuracy with 69.2%.

Table 7: Classification results for object recognition

<b>Bagging using Decision Tree Learner classifier</b>			
<i>Time taken to train model: 20.1 seconds</i>			
Class	Precision	Recall	F-Measure
Sofa	0.72	0.9	0.8
Chair	0.952	1	0.976
Table	0.938	0.75	0.833
Bin	0.75	0.9	0.818
Computer	0.909	1	0.952
Sink	0.692	0.692	0.692
Toilet	0.875	0.933	0.903
Car	0.857	0.947	0.9
Unknown	1	0.56	0.718
<b>All</b>	<b>0.865</b>	<b>0.849</b>	<b>0.844</b>

The confusion matrix is also shown in Table 8.

Table 8: Confusion matrix for object recognition

<b>Classified as</b> <b>Actual class</b>	Sofa	Chair	Table	Bin	Computer	Sink	Toilet	Car	Unknown
Sofa	18	0	1	0	0	0	0	1	0
Chair	0	20	0	0	0	0	0	0	0
Table	5	0	15	0	0	0	0	0	0
Bin	0	0	0	18	0	2	0	0	0
Computer	0	0	0	0	20	0	0	0	0
Sink	0	0	0	4	0	9	0	0	0
Toilet	0	1	0	0	0	0	14	0	0
Car	1	0	0	0	0	0	0	18	0
Unknown	1	0	0	2	2	2	2	2	14

Feature analysis is performed using chi-square feature selection techniques (Witten & Frank 2005). Chi-square feature ranking is a technique used to calculate the likelihood that a feature is correlated with a class. Based on the annotations in the corpus, this technique can estimate likelihoods per feature and rank the features that are most useful in the classification. This helps identify which features are important in the scene detection process.

Feature analysis is important especially in this case since we have added new features and are also using global features for each object and therefore the importance of these features in the classification process need to be analyzed.

The total number of features used for object recognition is 14951. Figure 24 shows the top 20 features in chi-square feature selection techniques.

```

=== Attribute Selection on all input data ===

Search Method:
    Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 14952 class):
    Chi-squared Ranking Filter

Ranked attributes:
733.5587172      3 length
722.565217      1 height
684.1997782     2 width
258.3158234     47 color36
219.7178308    2886 HOG2839
218.9285358    11660 HOG11613
216.5137668     8752 HOG8705
214.3137584     4210 HOG4163
203.8318187     1300 HOG1253
202.9287518    14607 HOG14560
199.6810056    11207 HOG11160
199.1839542      23 color12
198.515526      35 color24
197.5280366    11153 HOG11106
196.1015129      65 HOG18
195.4730311     7095 HOG7048
194.559309      5026 HOG4979
194.0941077     3701 HOG3654
193.82577       6647 HOG6600
193.398259     14609 HOG14562

```

Figure 24: Chi square feature ranking showing the top 20 features

The contribution of our work lies in showing that the new features that were added were the top three features for object recognition. The ranking of texture features are also shown in Table 9.

Table 9: Texture features ranking

<b>Rank</b>	<b>Attribute</b>
145.0657834	t_con_mean
130.3623349	t_e_mean
125.7927917	t_cor_std
91.6024273	t_con_std
84.0009182	t_H_std
80.5857029	t_e_std
76.7294235	t_H_mean

## CHAPTER 6- SCENE IDENTIFICATION

In the real world, a scene is a rich source of information that helps solve the challenging problems on image processing. Localization and tracking are applications of scene detection. By detecting the scene, information from visual elements and their surrounding is gathered. Because scenes can often be classified without full knowledge of every object in the image, the goal is not as ambitious as object recognition. For instance, if a person recognizes a chair is under a table, they may hypothesize that they are looking at a classroom scene, even if they cannot see every detail in the image. It may be possible in some cases to use low-level information, such as color or texture, to classify scene types accurately. This chapter focuses on automatic scene detection using global features with local representations to show the gradient structure of an image. Different machine learning approaches was applied and the system showed promising results. The output of this work counts as a contextual cueing and is used in object recognition. For this purpose, k nearest neighbor (kNN) classifier is used to calculate the likelihood of an image belonging to different scene classes.

By specifying the scene, global information of image is extracted which can help different processes in image understating, video segmenting, indexing and annotating. Knowing the scene class helps object recognition by decreasing the number of object classes, scales and positions that must be considered. Researchers who study scene detection fall in two categories. The first group tries to identify the scene by taking into consideration the type of objects it contains. This group of researchers looks to identify the important elements of an image and use that to detect the type of the scene. For example by identifying the sky or clouds the scene can be categorized

as outdoor, or by detecting walls in an image the scene can be classified as indoor (Li et al. 2009).

The second group use global features to identify the scene. The goal of this group is to view the entire image and use that information to identify the scene of the image. Global features are based on spatial layout properties and are not based on different segments of the image and therefore are not object oriented (Oliva and Torralba 2006; Mallepudi et al. 2011).

In this chapter the goal is to combine the two approaches by generating local gradients and global color and texture features to identify the scene. For this purpose, a dataset has been created that consists of images from 5 different scenes: classroom, computer lab, bathroom, corridor and outdoor.

## **6.1 Global Feature Vector**

All scene classifying systems extract appropriate features and use some sort of learning or pattern recognition engine to classify the image. Our approach is to understand gist based methods on scene-centered, rather than object-centered primitives. Global features are based on configurations of spatial scales and are estimated without invoking segmentation or grouping operations. By relying on low level feature detectors across large regions of the visual field, we can build a holistic and low dimensional representation of the structure of the scene. We use a framework of low-level features (multi-scale Gabor filters and color histogram), coupled with supervised learning to estimate the label for a scene.

An important approach for extracting texture features is using wavelet transforms. Wavelet transforms refer to the process of decomposition of a signal. Basically, Gabor filters are a group



of wavelets, with each wavelet capturing energy at a specific frequency and a specific direction (Zhang et al. 2000). Gabor filters have been used in image applications such as texture classification, object recognition, segmentation, content based image retrieval and motion tracking.

Four (4) scales and six (6) orientations make the Gabor filter useful for image processing. The mean and the standard deviation of the filtered images are used as features.

A case study by Li et al. evaluated the performance of texture descriptors using sample images of rocks. In their study, the authors found that Gabor filters outperformed other texture descriptors (Li et al. 1999).

By applying the Gabor filters to a given image a set of filtered images are produced. Each of the filters estimates the energy along a specific frequency and orientation of the input signal.

The discrete Gabor wavelet transforms for a given image  $I(x, y)$ , is obtained by a convolution (Chen et al. 2004) using Equation 17:

$$G_{mn}(x, y) = \sum_s \sum_t I(x-s, y-t) \Psi_{mn}^*(s, t) \quad (\text{Eq. 17})$$

where  $s$  and  $t$  are filter mask size variables,  $\Psi_{mn}^*$  is the complex conjugate of  $\Psi_{mn}$  is a class of self-similar functions generated from dilation and rotation of the following mother wavelet in Equation 18 (Chen et al. 2004):

$$\Psi(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \cdot \exp(j2\pi Wx) \quad (\text{Eq. 18})$$

where  $W$  is called the modulation frequency. The self-similar Gabor wavelets are obtained using Equation 19:

$$\Psi_{mn}(x, y) = a^{-m} \Psi(\tilde{x}, \tilde{y}) \quad (\text{Eq. 19})$$

where:

$$\tilde{x} = a^{-m} (x \cos \theta + y \sin \theta) \quad (\text{Eq. 20})$$

$$\tilde{y} = a^{-m} (-x \sin \theta + y \cos \theta)$$

where  $m=0,1,\dots,M-1$  specifies the scale and  $n=0,1,\dots,N-1$  specifies the orientation of the wavelet.  $M$  is the number of scales and  $N$  is the number of orientations.  $a$  is greater than 1 and represents the scale factor and is dependent on the higher center frequency and lower center frequency of interest and  $\theta = n\pi / N$  (Chen et al. 2004).

After applying the Gabor filters on an image, an array of magnitudes is obtained. This contains the means and standard deviations which represent the texture feature components.

Color is an important dimension of human visual perception as it helps in recognizing and discriminating visual content. Color features have been found to be effective for indexing and searching color images, and these features can be easily extracted and matched (Han and Ma 2002). The most common color metric used in the literature is the color histogram.

Each histogram bin is represented by a range of colors and the color histogram represents the coarse distribution of the colors in the image (Han and Ma 2002). So if two colors are located in the same bin they are treated as similar colors. On the other hand if two colors are located in different bins they are considered different, even if they might be very similar to each other.

By mapping the image to an appropriate color space, quantizing the mapped image and then counting the occurrence of each color, the color histogram for the image is obtained.

By using the color histogram, similarity of color features is specified by counting the color intensities. Any color is reproduced by combining the three primary colors (R, G and B) together. Therefore these three colors represent colors as vectors in 3D RGB color space.

The final features are Histogram of Gradient (HOG) (Dalal and Triggs 2005). For each image after normalizing color, indirect gradient is extracted for each cell. Each pixel within the cell votes for an oriented based histogram bin based on the values found in the gradient computation. For each cell 36 bins are specified. In this case after decomposing an image into 24 x24 pixels cell, HOG dimension is 3500. Figure 25 shows the HOG representation of an image taken form a corridor.

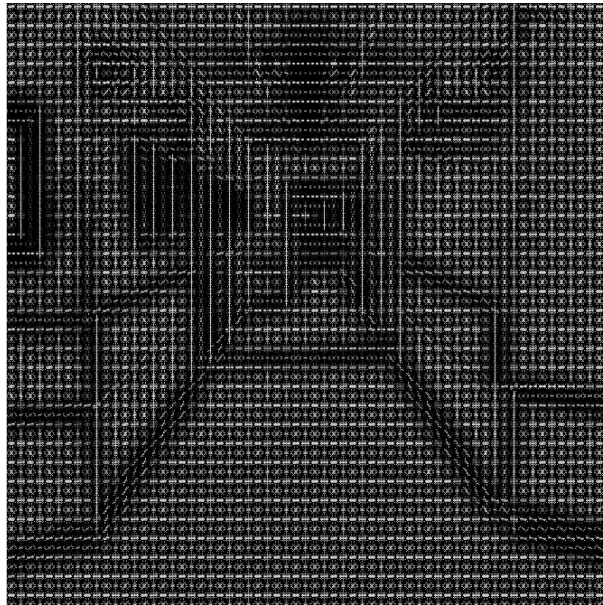


Figure 25: HOG representation for corridor

## 6.2 Scene Feature Analysis

A new set of features consisting local gradients and global color and texture features are extracted and used for scene recognition. In this section the importance of these features are studied by first testing the features on a dataset consisting scene images and then analyzing the features using Chi-square feature ranking technique.

WEKA (Waikato Environment for Knowledge Analysis) (Bouckaert et al. 2010) is used for applying classification methods such as Naïve Bayes, Support Vector Machines and Decision Trees. WEKA is a machine learning software with a collection of machine learning algorithms for data mining tasks.

After the features have been extracted, a model is trained to detect the scene of each image. In the classification process, LibSVM, Decision Tree, Naïve Bayes and bagging classifiers are used.

There are 5 scene classes: bathroom, classroom, corridor, computer lab and outdoor.

764 images are used that contain images from different scene classes. For the purpose of evaluation the corpus is divided into two sections: one section for training purposes and the other section for testing purposes. The training set consists of 80% of the corpus and testing consists of the remaining 20%. 10 Fold cross validation is performed to increase the accuracy of the results. Classification results using bagging classifier is shown in Table 10.

Table 10: Classification results for scene identification

<b>Bagging using Random Forest classifier</b>			
<i>Time taken to train model: 9.19 seconds</i>			
Class	Precision	Recall	F-Measure
Bathroom	0.599	0.667	0.631
Classroom	0.556	0.631	0.591
Corridor	0.664	0.536	0.593
Computer lab	0.540	0.487	0.512
Outdoor	0.739	0.768	0.753
<b>All</b>	<b>0.619</b>	<b>0.617</b>	<b>0.615</b>

The confusion matrix is also shown in Table 11.

Table 11: Confusion matrix

<b>Classified as</b> <b>Actual class</b>	Bathroom	Classroom	Corridor	Computer lab	Outdoor
Bathroom	100	10	26	7	7
Classroom	12	99	3	33	10
Corridor	35	13	81	13	9
Computer lab	14	42	8	75	15
Outdoor	6	14	4	11	116

By looking at the result and the confusion matrix, it can be seen that bath and corridor were mainly misclassified with each other, and classroom and computer lab were misclassified with each other. Outdoor class showed the highest accuracy of 76.8%.

Since we are using global images for each scene and the features consist of color and texture features, therefore images with the same correlation, contrast, homogeneity and same range of colors tend to get classified together.

The results show that using global features improves the performance of scene detection and the results are competitive with the state of art.

## **CHAPTER 7- EVIDENCE FUSION MODEL**

For combining the multiple classification models that we have in our system, an evidence fusion system is developed to give the final object identification result. In this chapter, the process of fusing the information from previous stages and using information from ConceptNet is discussed.

An assumption of classifier fusion is that the classifier algorithms are imperfect and therefore one way of enhancing the performance of the classification system is to construct multiple systems and then combine the results.

### **7.1 Scene Likelihood**

In this section, the scene detection algorithm (Section 6.1) is used as input to generate the likelihood of each image belonging to the scene classes.

Knowing the scene improves the process of object detection (Oliva and Torralba 2007). For example if the scene of an image is recognized and labeled as computer lab, then by using the information extracted from ConceptNet, the systems' attention diverts to objects such as computer, chair, board, and scientific instruments. In other words, visual attention can be guided using the knowledge from the scene. Knowing the scene class will certainly help identify the objects in the scene and affects the expected likelihood and location of the objects it contains. Using scene recognition method, the likelihood of each scene will be determined.

The k-nearest neighbors (kNN) rule is one of the oldest and simplest methods for pattern classification. The kNN rule classifies each unlabeled data by the dominant number of labels

among its k-nearest neighbors in the training set. Its performance thus depends crucially on the distance metric used to identify nearest neighbors.

kNN classification is a supervised classification method that uses the class label of the training data. It is also one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data.

During the training process the true class of the data is used for training the classifier and during testing the class of each test sample is predicted using the classifier. In this method the predicted class of test samples is set equal to the most frequent true class among the k nearest training samples.

For kNN we have a matrix with all our training dataset. We will also generate a labeling matrix for our training data. For classification the Euclidean distance between each testing data and all the training data is calculated. The distance is sorted from low to high and the number of classes occurred among the k nearest neighbors is calculated and the data is assigned to the class which has occurred the most. Classification of the images was performed using a kNN classifier. We used cross validation to find the best amount for k in this method. Therefore we used 10-fold validation. kNN with  $k=12$  shows the highest accuracy therefore is the best choice for kNN classifier method.

The system was made to classify the dataset images among the five scene classes. With this approach, an overall accuracy of 63% has been achieved (Table 12). When comparing the

individual class accuracy scores on the testing data set it was observed that the lab class had the highest accuracy of 90% followed by corridor with 79 % (Table 13).

Table 12: Classification Results for test Images

Method	Number of correctly classified images	Number of Incorrectly classified images	% Accuracy
<b>KNN (k=12)</b>	65	38	63.11

Table 13: Performance for each class

Class	Bathroom	Classroom	Computer Lab	Corridor	Outdoor
<b>Performance</b>	0.50	0.36	0.9	0.79	0.70

In Table 14 examples are shown for a couple of testing images and the likelihood of each image being classified in each class is specified.

Results show the success rate on outdoor, lab and corridor. It confirms that some classroom scenes are classified as lab with particularly high probabilities. In cases that there is misclassification between similar scenes, the scene with the second highest likelihood is the correct class.

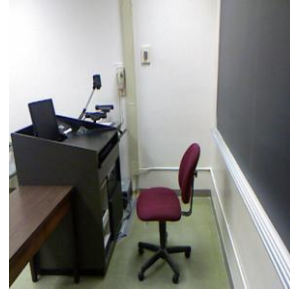
Therefore using the results from kNN in object recognition will decrease these errors.



Table 14: Test images with scene likelihoods



Bathroom: **0.6**  
Classroom: 0.32  
Corridor: 0  
Computer lab: 0.08  
Outdoor: 0



Bathroom: 0.12  
Classroom: 0.32  
Corridor: 0.06  
Computer lab: **0.46**  
Outdoor: 0.04



Bathroom: 0.16  
Classroom: **0.34**  
Corridor: 0.12  
Computer lab: 0.40  
Outdoor: 0



Bathroom: 0.20  
Classroom: **0.40**  
Corridor: 0.10  
Computer lab: 0.26  
Outdoor: 0.04



Bathroom: **0.32**  
Classroom: 0.22  
Corridor: 0.26  
Computer lab: 0.08  
Outdoor: 0.12



Bathroom: 0.16  
Classroom: 0.26  
Corridor: 0.02  
Computer lab: **0.50**  
Outdoor: 0.06



Bathroom: 0.10  
Classroom: 0.30  
Corridor: 0.08  
Computer lab: **0.32**  
Outdoor: 0.20




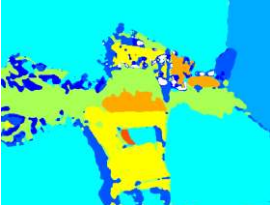
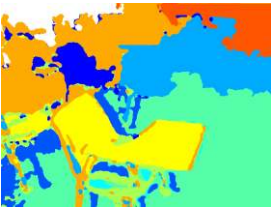


Bathroom: 0.18  
Classroom: 0.34  
Corridor: 0.08  
Computer lab: **0.36**  
Outdoor: 0.04

## 7.2 Object Likelihood

At this stage each image has been segmented to potential object regions. Object features for each segment are extracted. Using supervised classification method the likelihood of each segment belonging to each object class is generated.

kNN (k=10) is used as the classification method to generate the likelihood of each segment being classified in different object classes. For each image 3 regions with the highest number of embedded points have been selected and results are given in Table 15.

Table 15: Test images with object likelihoods

	<i>Object1(Cyan)</i> Bin: 0.4 Computer: 0.4 Unknown : 0.2		
	<i>Object1(Yellow)</i> Table: 0.2 Sofa: 0.3 Chair:0.3 Unknown: 0.2	<i>Object2(Blue)</i> Computer:0.1 Bin: 0.1 Sink:0.2 Unknown:0.5	<i>Object3(Green)</i> Computer: 0.2 Table: 0.3 Sofa: 0.4 Unknown: 0.1
	<i>Object1(Yellow)</i> Table: 0.2 Sofa: 0.3 Chair:0.4 Unknown: 0.1	<i>Object2(Orange)</i> Table: 0.2 Sofa: 0.4 Unknown: 0.4	<i>Object3(Blue)</i> Table: 0.2 Unknown: 0.8
	<i>Object1(Red)</i> Chair: 0.5 Computer: 0.2 Unknown: 0.2 Bin: 0.1	<i>Object2(Orange)</i> Chair: 0.4 Sofa:0.3 Computer: 0.1 Unknown: 0.1 Sink: 0.1	<i>Object3(Blue)</i> Computer: 0.4 Bin:0.4 Unknown: 0.2
	<i>Object1(Purple)</i> Computer: 0.5 Unknown: 0.3 Bin: 0.1 Table:0.1	<i>Object2(Red)</i> Table: 0.5 Unknown: 0.3 Bin: 0.1 Computer: 0.1	<i>Object3(Brown)</i> Unknown: 0.4 Computer: 0.3 Sink: 0.2 Bin: 0.1

### 7.3 ConceptNet Similarity

ConceptNet gathers commonsense knowledge thorough ordinary people in its site. The data is represented in the form of semantic network. A python processor is developed to access the ConceptNet database and extract the information needed to improve the process of image retrieval.

For selecting objects in each scenes, it is useful to know how related they are together. This can be identified by using ConceptNet and calculating the amount of activation that is spread from one concept to another (with a maximum of 1). In other words it shows how related the two concepts are together. Table 16 lists the activation measure for the objects and scenes.

Table 16: ConceptNet activation measure

<b>Scene class</b> <b>Object Class</b>	<b>Bathroom</b>	<b>Classroom</b>	<b>Computer Lab</b>	<b>Corridor</b>	<b>Outdoor</b>
<b>Sofa</b>	0.11514	0.28614	0.27047	0.42486	0.19021
<b>Chair</b>	0.35991	0.33169	0.22190	0.20738	0.10522
<b>Table</b>	0.30282	0.52157	0.39157	0.20357	0.10656
<b>Bin</b>	0.32614	0.24408	0.19240	0.11029	0.18119
<b>Computer</b>	0.08810	0.67433	0.65229	0.25383	0.08252
<b>Sink</b>	0.67572	0.24819	0.39216	0.24390	0.12233
<b>Toilet</b>	0.76891	0.07345	0.16543	0.27134	0.27464
<b>Car</b>	0.029772	0.03277	0.02151	0.09492	0.11278

### 7.4 Fusion Method and Results

At this stage the following components are calculated for each segment:

- Scene classes and their likelihood (e.g. Classroom: 0.6; Lab: 0.2)
- Object classes and their likelihood (e.g. Computer: 0.7; table: 0.15)
- Similarity measure between each scene and object using ConceptNet (e.g. 0.549)

A likelihood score is calculated by combining the scores from each of the above stages. The object is classified to the class with the highest score. The problem of finding the label for scene and object is like computing a probability for a sequence of events is observed in the world. Based on Equation 22, the three factors are multiplied to compute the score for each object. The most probable path is selected and the path shows the label of the object.

$$L(i) = \max_{i=1,\dots,N} LO_i * a_{ij} * LS_j, \quad \sum_{j=1}^M a_{ij} = 1 \quad \forall i \quad (\text{Eq. 22})$$

where  $LO_i$  is the state observation likelihood showing the likelihood of object class  $i$ ,  $LS_j$  is the state observation likelihood showing the likelihood of scene class  $j$ , and  $a_{ij}$  is the transition measure, each  $a_{ij}$  represents the likelihood of object class  $i$  existing in object scene  $j$  (Figure 26).

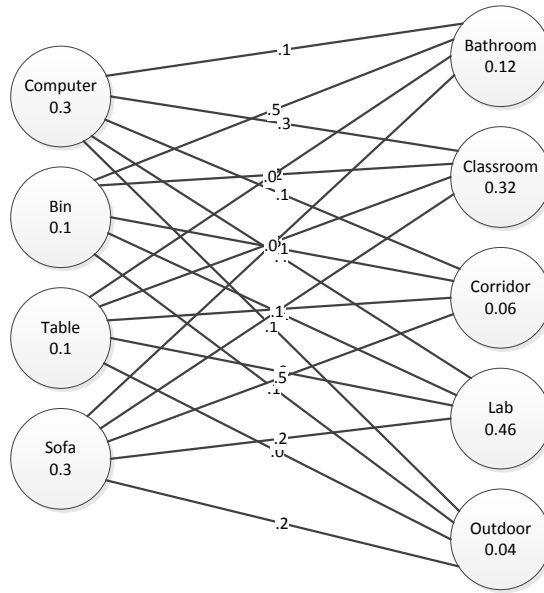




Figure 26: Visualizing the computation of object label

The overall idea of this approach is that there are multiple sources of information that provide partial classification. These classifications are joined in a way to give better final decision than any component classifier. This approach reduces the errors caused in different stages of the system. Obtaining full-scale image understanding in construction environments is our goal and we want to achieve this knowledge by taking into consideration the information from scene and objects together. Table 17 shows the final object recognition result for a sample image.

Table 17: Object recognition result for a sample image

		<i>Object1</i> Bin: 0.4 Computer: 0.4 Unknown : 0.2	Bathroom: <b>0.6</b> Classroom: 0.32 Computer lab: 0.08	$a_{Bin,Class} = 0.32005$ $a_{Bin,Bath} = 0.42766$ $a_{Bin,Lab} = 0.25229$ $a_{Computer,Class} = 0.47665$ $a_{Computer,Bath} = 0.06227$ $a_{Computer,Lab} = 0.46107$
$L(i) = \max_{i=1,...,N} LO_i * a_{ij} * LS_j = \max (0.4*0.6*0.42766, 0.4*0.32*0.32005, 0.4*0.08*0.25229, 0.4*0.6*0.06227, 0.4*0.32*0.47665, 0.4*0.08*0.46107 = 0.1026$ <p>So Object is “Bin” and scene is “Bathroom” for this image</p>				

The system’s performance on the sample test data was satisfactory. In the next stage of analyzing the framework images are taken from a room with more complex objects. The selected room has combination of occlusion (table blocking the chairs), illumination and shadow challenges.

We took two pictures from the same location of a room but with different objects. Figure 19 shows the original images and their segmentations. The system was able to identify the structures and therefore was able to interpret the scene. The chairs that are blocked by the table and only the back part of them were visible, were segmented together but in the object recognition process they were not recognized as chairs. But the chairs in the second picture were identified in the object recognition process and the system was able to handle illumination challenges.

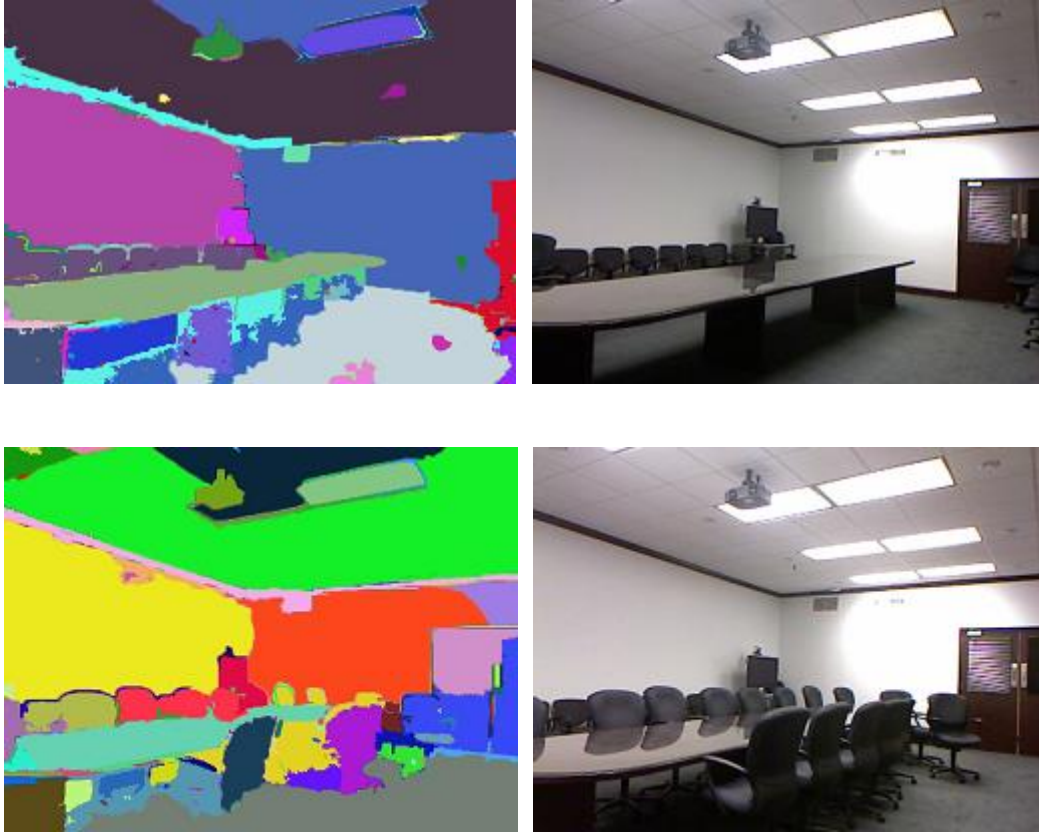


Figure 27: Testing the performance of the system for occlusion and illumination challenges

## 7.5 Text Generation

As explained in Chapter 2, linking individual words to images has a rich history. Researchers have been trying to predict words from image regions and build a sentence. Sentences are richer than lists of words, because they describe the most important objects, properties of objects, and relation between entities (among other things). In this part the semantic representations are converted into human readable and query-able natural language descriptions.

Once the classes for the scene and object are assigned, text generation system takes the information and automatically generates a text description. ConceptNet is also used to generate more descriptive sentences.



There are different semantic relations in ConceptNet embedded in different categories (a list of complete categories and relations is given in Table 2, Section 2.9). An example is given in Figures 28 shows different assertions that can be extracted from ConceptNet.

<pre>&gt;&gt;&gt; getassertions('chair') AtLocation(chair, office)[] AtLocation(chair, desk)[] IsA(chair, sit)[] HasProperty(chair, comfortable)[] HasPrerequisite(sit chair, chair)[] AtLocation(something, chair)[] AtLocation(cat, chair)[] UsedFor(chair, sit)[] AtLocation(chair, cubicle)[] AtLocation(chair, room)[] ['office', &lt;Assertion: AtLocation(chair, office)[]&gt;, 'desk', &lt;Assertion: AtLocation(chair, desk)[]&gt;, 'chair', &lt;Assertion: AtLocation(something, chair)[]&gt;, 'chair', &lt;Assertion: AtLocation(cat, chair)[]&gt;, 'cubicle', &lt;Assertion: AtLocation(chair, cubicle)[]&gt;, 'room', &lt;Assertion: AtLocation(chair, room)[]&gt;]</pre>	<pre>&gt;&gt;&gt; getassertions('computer') AtLocation(computer, office)[] PartOf(keyboard, computer)[] AtLocation(computer, library)[] PartOf(monitor, computer)[] AtLocation(computer, house)[] IsA(computer, electronic device)[] UsedFor(computer, play game)[] AtLocation(motherboard, computer)[] UsedFor(computer, work)[] PartOf(cpu, computer)[] ['office', &lt;Assertion: AtLocation(computer, office)[]&gt;, 'library', &lt;Assertion: AtLocation(computer, library)[]&gt;, 'house', &lt;Assertion: AtLocation(computer, house)[]&gt;, 'computer', &lt;Assertion: AtLocation(motherboard, computer)[]&gt;]</pre>
--	---

Figure 28: Assertions extracted from ConceptNet for ‘Chair’ and ‘Computer’

Once objects are detected in an image, the methodology automatically enriches the content by creating a new XML mark-up file with new metadata using the extracted information from the image. The detected objects and corresponding attributes are added to the file. Figure 29 presents an example of the XML mark-up. This mark-up approach can be used for automatic image to text processing.

```

<?xml version="1.0" encoding="ANSI_X3.4-1968"?>
  <image format="JPEG">
    <semantic>
      <Scene>
        <Scene name="Computer Lab"/>
      </Scene>
      <Objects_List>
        <Object id="01">
          <Name>Computer</Name>
          <IsA>electronic device</IsA>
          <UsedFor>Work or play game</UsedFor>
          <PartOf>Keyboard</PartOf>
          <PartOf>cpu </PartOf>
          <PartOf>monitor </PartOf>
          <Size Height="0.69 " Width="0.36 " Length="0.59"
            Units="Meter">
          </Size >
          <Color>Black</Color>
          <Position>Top Left</Position>
        </Object>
        ...
      </Objects_List>
      ...
    </semantic>
  </image>

```

Figure 29: XML format enrichment for an image



## **CHAPTER 8- CONCLUSION AND FUTURE WORK**

In this study, an automatic object and scene identification method has been developed, which bridges the semantic gap between low level features of image content and high level conceptual meaning. This led to a structural framework for organizing information (ontology) that generates text descriptions in natural language, based on understanding of image content.

We developed a new system that used contextual and common sense information for improving object recognition and scene detection. Information of scene and objects were fused together to reduce the level of uncertainty. This study in addition to improving segmentation, scene detection and object recognition can be used in different applications such as reverse engineering, marketing applications and computer-based image retrieval (CBIR).

The work in this research aimed to improve the task of image understanding and have a positive impact on tasks such as object recognition, scene detection and segmentation processes. The result of this dissertation can be used in applications that require physical parsing of the image into objects, surfaces and their relations. The applications include robotics, social networking, intelligence and anti-terrorism efforts, criminal investigations and security, marketing, and building information modeling in the construction industry. In this dissertation a structural framework (ontology) is developed that generates text descriptions in natural language, based on understanding of objects, structure and the attributes of an image.

The important contributions of this work include:

- Creating an annotated dataset for training and testing purposes which consist of images with labeled scene and objects.

- Developing a new method for segmenting image by integrating graph-based segmentation, surface index and feature vector for each pixel.
- Developing a new set of features for object recognition.
- Developing a new scene detection model by integrating local gradients and global color and texture features.
- Developing an evidence fusion model combining scene detection, object detection findings and commonsense knowledge.

## **8. 1 Recommendations for Future Work**

Future work will focus on incorporating video to be able to take advantage of multi-view geometry. Rules in video processing help to distinguish objects from their backgrounds and will give more information by capturing data from different points of view. Using multi-view imaging mimics how humans capture information by moving their head and looking at an object from different points of view. By having more 3D points, the combination of 3D interest points and HOG can be calculated which is called 3DHOG. Extension of HOG feature will overcome problem of variable viewpoints and partial occlusion.

Future work will also focus on improving the filters used in the post segmentation process that will result in rigid segments and remove tiny polygons and distortions that are not part of the object.

Objects may profoundly change when they participate in relations (e.g. the property of chair being occluded differs from a chair in a stable position). Grammar based models generalize deformable part models by representing objects using variable hierarchical structures. Each part

in a grammar-based model can be defined directly or in terms of other parts. Each model consists of a set of rules that define how the objects can be broken down into parts. Face recognition models is the main research done in shape grammar in content image retrieval. Future work can be extending this idea to all of the objects which can be defined by parts and set of rules for different combination of parts (geometric, appearance, etc.). This model will allow a representation of conditional independencies between parts and also will define a probabilistic distribution over all combination of parts for each object. The challenging part is finding relation between parts which is bond and connections (geometrical rules), joints and junctions (collinear, parallel or symmetric relations) or semantic relations (can gather information from commonsense knowledge sources).

Semantic relations and specifically spatial relationship between objects and structures are another topic of future research. These relations provide structural knowledge and can be applied for recognizing human/object interactions.

## REFERENCES

- E. Arias-Castro and D. L. Donoho, "Does median filtering truly preserve edges better than linear filtering?," *The Annals of Statistics*, pp. 1172-1206, 2009.
- C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley FrameNet project," in *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, 1998, pp. 86-90.
- K. Barnard, P. Duygulu, and D. Forsyth, "Clustering art," in *In IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, 2001, pp. 434-441 vol. 2.
- C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, pp. 121-167, 1998.
- N. Burrus, "Object Modeling and Detection," in *Hacking the Kinect*, J. Kramer, N. Burrus, D. Herrera C., F. Echtler, and M. Parker, Eds., ed, 2012.
- R. A. Calix, "Automated Semantic Understanding of Human Emotions in Writing and Speech," PhD, Engineering Science, Louisiana State University, 2011.
- J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on*, pp. 679-698, 1986.
- C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on*, vol. 24, pp. 1026-1038, 2002.
- C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *National Taiwan University, Computer Science and Information Engineering*, 2001.
- R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proceedings of the IEEE*, vol. 83, pp. 705-741, 1995.
- C. H. Chen, L. Potdat, and R. Chittineni, "Two Novel ACM (active Contour Model) Methods for Intravascular Ultrasound Image Segmentation," in *AIP*, 2010.
- L. Chen, G. Lu, and D. Zhang, "Effects of different Gabor filters parameters on image retrieval by texture," in *10th International Multimedia Modelling Conference*, 2004, pp. 273-278.
- Y. Chen, J. Z. Wang, and R. Krovetz, "Clue: Cluster-based retrieval of images by unsupervised learning," *IEEE Transactions on Image Processing*, vol. 14, pp. 1187-1201, 2005.
- R. S. Choras, "Image feature extraction techniques and their applications for CBIR and biometrics systems," *International Journal of Biology and Biomedical Engineering*, vol. 1, pp. 6-16, 2007.

- C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273-297, 1995.
- D. Crandall, P. Felzenszwalb, and D. Huttenlocher, "Spatial priors for part-based recognition using statistical models," in *Computer Vision and Pattern Recognition*, 2005, pp. 10-17 vol. 1.
- D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg, "Mapping the world's photos," in *18th international conference on World Wide Web*, 2009, pp. 761-770.
- N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition*, 2005, pp. 886-893 vol. 1.
- J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248-255.
- Y. Deng and B. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 800-810, 2001.
- C. Desai, D. Ramanan, and C. Fowlkes, "Discriminative models for multi-class object layout," in *ECCV'10 Proceedings of the 11th European conference on Computer vision: Part IV*, 2009, pp. 229-236.
- H. Du, P. Henry, X. Ren, M. Cheng, D. B. Goldman, S. M. Seitz, and D. Fox, "Interactive 3D modeling of indoor environments with a consumer depth camera," in *Proceedings of the 13th international conference on Ubiquitous computing*, 2011, pp. 75-84.
- R. M. Dufour, E. L. Miller, and N. P. Galatsanos, "Template matching based object recognition with unknown geometric parameters," *IEEE Transactions on Image Processing*, vol. 11, pp. 1385-1396, 2002.
- J. a. G. Eakins, M, "Content-based image retrieval, University of Northumbria at Newcastle," *JISC Technology Applications*, 1999.
- A. Eguchi and C. Thompson, "Object Recognition Based on Shape and Function: Inspired by Children's Word Acquisition," *Undergraduate Research*, vol. 13, pp. 38-49, 2012.
- M. a. Z. Everingham, A. and Williams, C. K. I. and VanGool, L. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results [Online]. Available: <http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf>
- A. Farhadi, M. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, "Every picture tells a story: Generating sentences from images," *Computer Vision–ECCV 2010*, pp. 15-29, 2010.

- P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1627-1645, 2010.
- P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, pp. 167-181, 2004.
- P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, pp. 55-79, 2005.
- H. M. Feng and T. S. Chua, "A bootstrapping approach to annotating large image collection," in *5th ACM SIGMM international workshop on Multimedia information retrieval 2003*, pp. 55-62.
- M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381-395, 1981.
- M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on Computers*, vol. 100, pp. 67-92, 1973.
- A. C. Gallagher, "Using Vanishing Points to Correct Camera Rotation in Images," presented at the *Second Canadian Conference on Computer and Robot Vision*, 2005.
- R. X. Gao, T. F. Wu, S. C. Zhu, and N. Sang, "Bayesian inference for layer representation with mixed markov random field," in *LNCS*, 2007, pp. 213-224.
- M. Golparvar-Fard, S. Savarese, and F. Peña-Mora, "Automated Model-Based Recognition of Progress Using Daily Construction Photographs and IFC-Based 4D Models," in *Construction Research Congress*, 2010.
- S. Gould, T. Gao, and D. Koller, "Region-based segmentation and object detection," ed: *Neural Information Processing Systems*, 2009, pp. 655-663.
- H. Grabner, J. Gall, and L. Van Gool, "What makes a chair a chair?," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1529-1536.
- C. Gu, J. J. Lim, P. Arbeláez, and J. Malik, "Recognition using regions," in *Computer Vision and Pattern Recognition*, 2009, pp. 1030-1037.
- E. Guldogan and M. Gabbouj, "Feature selection for content-based image retrieval," *Signal, Image and Video Processing*, vol. 2, pp. 241-250, 2008.
- A. Gupta and L. Davis, "Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers," *Computer Vision—ECCV 2008*, pp. 16-29, 2008.
- J. Han and K.-K. Ma, "Fuzzy color histogram and its use in color image retrieval," *IEEE Transactions on Image Processing*, vol. 11, pp. 944-952, 2002.

- C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision*, 1988, pp. 147-151.
- R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*: Cambridge University Press, 2000.
- C. Havasi, R. Speer, and J. Alonso, "ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge," in *Recent Advances in Natural Language Processing*, 2007, pp. 27-29.
- P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments," in the *12th International Symposium on Experimental Robotics (ISER)*, 2010, pp. 22-25.
- D. Hoiem, A. A. Efros, and M. Hebert, "Recovering occlusion boundaries from an image," *International Journal of Computer Vision*, pp. 1-19, 2011.
- D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using RGB-D cameras," in *RoboCup Symposium*, 2011, pp. 306-317.
- P. V. C. Hough, "Methods and Means for Recognizing Complex Patterns," *United States Patent*, 1962.
- T. Kadir, D. Boukerroui, and M. Brady, "An analysis of the scale saliency algorithm," *OUEL* No: 2264, vol. 3, 2003.
- T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," *Computer Vision-ECC*, pp. 228-241, 2004.
- M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321-331, 1988.
- K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, pp. 1437-1454, 2012.
- A. Kolb, E. Barth, R. Koch, and R. Larsen, "Time-of-Flight Cameras in Computer Graphics," in *Computer Graphics Forum*, 2010, pp. 141-159.
- J. Košecká and W. Zhang, "Video compass," in *Computer Vision—ECCV 2002*, ed: Springer, 2006, pp. 476-490.
- K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1817-1824.
- K. Lai and D. Fox, "Object recognition in 3D point clouds using web data and domain adaptation," *The International Journal of Robotics Research*, vol. 29, pp. 1019-1037, 2010.

- J. Li and J. Z. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1075-1088, 2003.
- J. Li and J. Z. Wang, "Real-time computerized annotation of pictures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 985-1002, 2008.
- L. J. Li, R. Socher, and L. Fei-Fei, "Towards total scene understanding: Classification, annotation and segmentation in an automatic framework," in *CVPR*, 2009, pp. 2036-2043.
- P. Lipson, E. Grimson, and P. Sinha, "Configuration based scene classification and image indexing," in *CVPR*, 1997, pp. 1007-1013.
- C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing: Label transfer via dense scene alignment," in *CVPR*, 2009, pp. 1972-1979.
- Y. Liu, D. Zhang, G. Lu, and W. Y. Ma, "Region-based image retrieval with perceptual colors," *Advances in Multimedia Information Processing-PCM 2004*, pp. 931-938, 2005.
- Y. Liu, D. Zhang, G. Lu, and W. Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognition*, vol. 40, pp. 262-282, 2007.
- L. C. Loschky, A. Sethi, D. J. Simons, T. N. Pydimarri, D. Ochs, and J. L. Corbeille, "The importance of information localization in scene gist recognition," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 33, p. 1431, 2007.
- D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision*, 1999, pp. 1150-1157 vol. 2.
- D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91-110, 2004.
- R. Lukac, B. Smolka, K. Martin, K. N. Plataniotis, and A. N. Venetsanopoulos, "Vector filtering for color imaging," *Signal Processing Magazine, IEEE*, vol. 22, pp. 74-86, 2005.
- A. M. Lytle, "A Framework for Object Recognition in Construction Using Building Information Modeling and High Frame Rate 3D Imaging," *Virginia Polytechnic Institute and State University*, 2011.
- J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," in *5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281-297.
- S. A. Mallepudi, R. A. Calix, and G. M. Knapp, "Material classification and automatic content enrichment of images using supervised learning and knowledge bases," in *SPIE 7881*, San Francisco 2011.



- J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, pp. 761-767, 2004.
- R. Mehrotra and J. E. Gary, "Similar-shape retrieval in shape data management," *Computer*, vol. 28, pp. 57-62, 1995.
- V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "An ontology approach to object-based image retrieval," in *ICIP*, 2003, pp. 511-514.
- Microsoft Kinect. (2013). Available: <http://www.xbox.com/en-US/kinect>
- K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Eighth International Conf. Computer Vision*, 2001, pp. 525-531 vol. 1.
- K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," *Computer Vision-ECCV 2002*, pp. 128-142, 2002.
- G. A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, pp. 39-41, 1995.
- P. Mulhem<sup>Y</sup>, W. K. Leow<sup>P</sup>, and Y. K. Lee<sup>P</sup>, "Fuzzy conceptual graphs for matching images of natural scenes," To appear in *IJCAI*, p. 1, 2001.
- C. D. Mutto, P. Zanuttigh, and G. M. Cortelazzo, *Time-of-Flight Cameras and Microsoft Kinect (TM)*, 2012.
- R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011, pp. 127-136.
- A. Oliva and A. Torralba, "Building the gist of a scene: The role of global image features in recognition," *Progress in brain research*, vol. 155, pp. 23-36, 2006.
- A. Oliva and A. Torralba, "The role of context in object recognition," *Trends in cognitive sciences*, vol. 11, pp. 520-527, 2007.
- K. N. Plataniotis and A. N. Venetsanopoulos, *Color image processing and applications*: Springer Verlag Wien, 2000.
- K. Porkaew, M. Ortega, and S. Mehrotra, "Query reformulation for content based multimedia retrieval in MARS," in *Multimedia Computing and Systems 1999*, pp. 747-751.
- A. Quattoni, M. Collins, and T. Darrell, "Conditional random fields for object recognition," in *Neural Information Processing Systems*, 2004.

- L. Rabiner and B. Juang, "An introduction to hidden Markov models," *ASSP Magazine, IEEE*, vol. 3, pp. 4-16, 1986.
- N. Rafibakhsh, J. Gong, M. Siddiqui, C. Gordon, and H. F. Lee, "Analysis of XBOX Kinect Sensor Data for Use on Construction Sites: Depth Accuracy and Sensor Interference Assessment," in *Constitution Research Congress*, 2012, pp. 848-857.
- A. Richtsfeld and M. Vincze, "3D Shape Detection for Mobile Robot Learning," in *Advances in Robotics Research*, ed: Springer, 2009, pp. 99-109.
- E. Rivlin, S. J. Dickinson, and A. Rosenfeld, "Recognition by functional parts " in *Comp. Vision and Img. Understanding*, 1995, pp. 164-176.
- P. M. Roth and M. Winter, "Survey of appearance-based methods for object recognition," *Inst. for Computer Graphics and Vision, Graz University of Technology, Austria, Tech. Rep. ICG-TR-01/08*, 2008.
- C. Rother, "A new approach to vanishing point detection in architectural environments," *Image and Vision Computing*, vol. 20, pp. 647-655, 2002.
- Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool for interactive content-based image retrieval," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 8, pp. 644-655, 1998.
- B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: a database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, pp. 157-173, 2008.
- M. A. Sadeghi and A. Farhadi, "Recognition using visual phrases," in *Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1745-1752.
- B. Scholz-Reiter, H. Thamer, and C. Uriarte, "An Approach for 3D Object Recognition of Universal Goods," *International Journal of Computers*, vol. 5, 2011.
- A. Shah-hosseini, "Semantic Image Retrieval Using Relevance Feedback and Transaction Logs," *Louisiana State University*, 2007.
- J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888-905, 2000.
- C. R. Shyu, C. E. Brodley, A. C. Kak, A. Kosaka, A. M. Aisen, and L. S. Broderick, "ASSERT: a physician-in-the-loop content-based retrieval system for HRCT image databases," *Computer Vision and Image Understanding*, vol. 75, pp. 111-132, 1999.
- N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," presented at the *European Conference on Computer Vision(ECCV)*, 2012.

- W. C. Siu and H. J. Zhang, Multimedia information retrieval and management: Technological fundamentals and applications: Springer Verlag, 2003.
- J. R. Smith and S. F. Chang, "Automated image retrieval using color and texture," IEEE Transaction on Pattern Analysis and Machine Intelligence, 1996.
- R. Speer, C. Havasi, and H. Lieberman, "AnalogySpace: Reducing the dimensionality of common sense knowledge," in Proceedings of AAAI, 2008.
- G. Srinivasan and G. Shobha, "Statistical texture analysis," proceedings of world academy of science, engg & tech, vol. 36, 2008.
- J. K. Steeves, G. K. Humphrey, J. C. Culham, R. S. Menon, A. D. Milner, and M. A. Goodale, "Behavioral and neuroimaging evidence for a contribution of color and texture information to scene classification in a patient with visual form agnosia," Journal of Cognitive Neuroscience, vol. 16, pp. 955-965, 2004.
- F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in Proceedings of the 16th international conference on World Wide Web, 2007, pp. 697-706.
- B. Sumengen, B. Manjunath, and C. Kenney, "Image segmentation using multi-region stability and edge strength," in International Conference on Image Processing (ICIP), , Barcelona, 2003, pp. 429-32 vol. 2.
- J. Y. Sun, Z. X. Sun, R. H. Zhou, and H. F. Wang, "A semantic-based image retrieval system: VisEngine," in 1st Int. Conf on Machine Learning and Cybernetics, 2002, pp. 349-353 vol. 1.
- H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," IEEE Transactions on Systems, Man and Cybernetics, vol. 8, pp. 460-473, 1978.
- P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle, "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques," Automation in Construction, vol. 19, pp. 829-843, 2010.
- J. F. Tasič, M. Najim, and M. Ansorge, Intelligent integrated media communication techniques: cost 254 & cost 276: Springer Netherlands, 2003.
- S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in ninth ACM international conference on Multimedia, 2001, pp. 107-118.
- C. Town and D. Sinclair, "Content based image retrieval using semantic visual categories," AT&T Laboratories Cambridge Report, 2000.
- L. Trujillo and G. Olague, "Automated design of image operators that detect interest points," Evolutionary Computation, vol. 16, pp. 483-507, 2008.

- T. Tuytelaars and L. Van Gool, "Matching widely separated views based on affine invariant regions," *International Journal of Computer Vision*, vol. 59, pp. 61-85, 2004.
- A. Vailaya, M. A. T. Figueiredo, A. K. Jain, and H. J. Zhang, "Image classification for content-based indexing," *IEEE Transactions on Image Processing*, vol. 10, pp. 117-130, 2001.
- M. E. L. Van and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int J Con1putVis*, vol. 88, pp. 3-338, 2010.
- N. Vasconcelos and A. Lippman, "Library-based coding: A representation for efficient video compression and retrieval," in *Data Compression Conference (DCC97)*, 1997, pp. 121-130.
- J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 947-963, 2001.
- J. Webb and J. Ashley, *Beginning Kinect Programming with the Microsoft Kinect SDK*: Apress, 2012.
- I. T. Weerasinghe, J. Y. Ruwanpura, J. E. Boyd, and A. F. Habib, "Application of Microsoft Kinect sensor for tracking construction workers," in *Construction Research Congress 2012@sConstruction Challenges in a Flat World*, 2012, pp. 858-867.
- I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*: Morgan Kaufmann, 2005.
- Y. Wu and A. Zhang, "A feature re-weighting approach for relevance feedback in image retrieval," in *Image Processing ICIP*, 2002, pp. 581-584 vol. 2.
- B. Yao, X. Yang, and S. Zhu, "Introduction to a large-scale general purpose ground truth database: methodology, annotation tool and benchmarks," *Lecture Notes in Computer Science*, vol. 4679, p. 169, 2007.
- B. Z. Yao, X. Yang, L. Lin, M. W. Lee, and S. C. Zhu, "I2t: Image parsing to text description," *Proceedings of the IEEE*, vol. 98, pp. 1485-1508, 2010.
- C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Transactions on Computers*, vol. 100, pp. 68-86, 1971.
- D. Zhang, A. Wong, M. Indrawan, and G. Lu, "Content-based image retrieval using Gabor texture features," in *IEEE Pacific-Rim Conference on Multimedia*, University of Sydney, Australia, 2000.
- X. Zhang, N. Bakis, T. C. Lukins, Y. M. Ibrahim, S. Wu, M. Kagioglou, G. Aouad, A. P. Kaka, and E. Trucco, "Automating progress measurement of construction projects," *Automation in Construction*, vol. 18, pp. 294-301, 2009.

## **VITA**

Mehdi Khazaeli received his bachelor of science at Isfahan University of Technology in Industrial Engineering in 2006. Thereafter, he went to Liverpool to study Masters and received his degree in Product Design and Management at University of Liverpool in 2009 under supervision of Dr. Hugh Clare. In January of 2010, he started graduate studies in the college of engineering at Louisiana State University (LSU) as a teaching assistant in Industrial Engineering program. During his years at LSU he has worked as an instructor and teacher assistant for the Industrial Engineering program.

He is a candidate for the Doctor of Philosophy degree in Engineering Science with concentration in Information Technology and Engineering (ITE) under supervision of Dr. Gerald Knapp. The degree will be conferred at the summer commencement 2013.