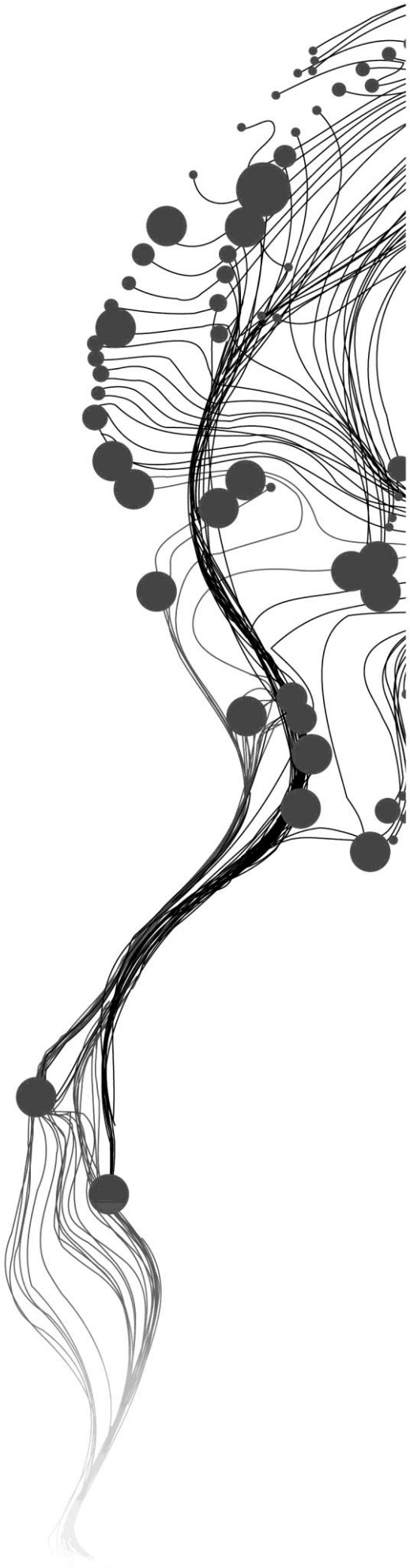
# RECOGNITION OF OBJECTS IN RGB-D DATA OF BUILDING INTERIORS

KEZHEN LI
February, 2013

SUPERVISORS:
Dr. K. Khoshelham
Dr.Ir. S.J. Oude Elberink



# RECOGNITION OF OBJECTS IN RGB-D DATA OF BUILDING INTERIORS

KEZHEN LI
Enschede, The Netherlands, February, 2013

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Geoinformatics

SUPERVISORS:
Dr. K. Khoshelham
Dr.Ir. S.J. Oude Elberink

THESIS ASSESSMENT BOARD:
Prof.Dr.Ir. M.G. Vosselman (Chair)
Dr. R.C. Lindenbergh, (External Examiner, Delft University of Technology)
Prof. Ma Zhimin, Chang'an University

# ABSTRACT

With more and more demand for accurate indoor model, accurate 3d indoor reconstruction and recognition is needed. In order to build the indoor models, the first procedure is to recognize walls, floors, ceilings and objects of indoor environments in the acquired data. Acquired by the emerging new sensor Kinect, the RGB-D data can be divided into RGB color image and depth image .Our research aims at taking advantage of both RGB and depth to develop an effective and robust methodology to recognize indoor objects using the RGB-D data acquired by Kinect.

In this research, a feature based recognition and classifier training method are combined. First step, gradient descriptor and size descriptor are adopted to extract corresponding local features from RGB, depth and point clouds of the training data. A large dataset with multi-view of objects are used as the training data. Those local features are then quantized and clustered using the bag of words concept. A histogram which summarizes the local features of each training image is later generated. The visual words of the cluster are made for further use. After then, using those features over the training data, the SVM classifier and QDC classifier are trained for discriminating different classes of objects. While on the other hand, we apply the same feature extraction using the visual words made previously to our test image. The recognition is conducted by applying the trained classifier to the test features. Two recognition experiments are carried out in our work: 1. self test recognition. 2. Recognition of data of real scene which contain multiple objects and background.

Conclusion is drawn from the experiment and the recognition result that using SVM as the training classifier, the combination of the features from RGB and depth data has a superior performance over the individual one.

**Keywords**: object recognition, RGB-D data, gradient feature, size feature, bag of words, SVM classifier

# ACKNOWLEDGEMENTS

It is a great opportunity to conclude my MSc. and express my gratitude to all the people who has supported me. The Study time in ITC is intensive but worthy. It is little hard to list all the names and people who had helped me to make this achievements but I will try.

First and foremost, I would like to express my greatest gratitude and deep regards to my first supervisor Dr Khoshelham, who had gave me continuous support and brilliant suggestions during this research. It is you who proposed such an innovative and interesting research topic and taught me immense knowledge of this field. This work would not have been possible without your patient attitude and clear guidance in all the time of research and writing of this thesis. I also would like to offer my special thanks to my second supervisor Dr.Ir. S.J. Oude Elberink, for your inspiring ideas and a great help in the improvements in the document. I am also honoured to receive lectures from Prof.Dr.Ir. M.G. Vosselman in laser scanning and image processing, which make me interested in this field.

My special thanks are extended to the PHD and staff of EOS department, Daniel, Biao Xiong, Liang Zhou, Sudan Xu and Jinfeng Mu, for their advice and supports to my research.

I would also like to thank the professors and lecturers in the GFM programme, except the huge knowledge and technique you taught me .Your immense knowledge and critical scientific attitude helped me to cultivate a scientific critical thinking and develop a strong interest in Geoinformatics field.

It is a valuable and unforgettable experience to take the GFM course and have a great friendship with so many kind people from all over the world: Manuel, Arun, Mervat, Amare, Razan, Cyprian, Bezaye, Albert, Bayarmaa, Federico, Dan, Mina, Frodouald, Parya, Cheng, Fatma, Andreas. I will never forget the time we spent together: taking course, enjoying party, nice excursion and travelling experience.

I also want to give my thanks to all the people who supported me and had precious moments with me during the Msc study in ITC. The fellows from china, I feel the warm of home with your concern and support. The friend from all over the world, our communication of cross culture has broadened my horizon as well as introduced the Chinese culture to the world.

At last, I offer my greatest appreciation to my mom, dad for their infinite support in everything. Your son has made some work in research.

# TABLE OF CONTENTS

List of figures ..... iv
List of tables ..... vi
1. Introduction ..... 1
1.1. Motivation and problem statment ..... 1
1.2. Research identification ..... 2
1.3. Thesis sturcture ..... 3
2. Literature review ..... 5
2.1. Characterization of the Microsoft Kinect ..... 5
2.2. Indoor mapping and modeling using RGB-D data ..... 6
2.3. Obeject recognition by using Feature descriptors ..... 7
2.4. Classification or matching method ..... 10
2.5. Recognition over the images from real world scene ..... 12
2.6. Summary ..... 12
3. Research methodology ..... 13
3.1. Introduction ..... 13
3.2. Framework of the methodology ..... 13
3.3. Local Feature extraction ..... 14
3.4. Aggregating local features using the Bag-of-words concept ..... 18
3.5. Classifier training phase: ..... 20
3.6. Self test recognition ..... 22
3.7. Recognition over images of real scene ..... 23
3.8. Visualization: ..... 24
3.9. Quality assessment ..... 24
4. Feature extraction and recognition results ..... 25
4.1. Introduction ..... 25
4.2. Training dataset ..... 25
4.3. Local feature extraction ..... 27
4.4. Generation of bag of words ..... 33
4.5. Classifier training ..... 39
4.6. Recognition over images of real scene ..... 39
5. Quality Assessment and discussion ..... 45
5.1. Self test recognition assessment and analysis ..... 45
5.2. Assessment over recognition image of scene ..... 48
6. Conclusion and recommendations ..... 51
6.1. Conclusion ..... 51
6.2. Answers to the research questions: ..... 52
6.3. Recommendations ..... 53
List of references ..... 55

# LIST OF FIGURES

Figure 2-1:Kinect....5
Figure 2-2: Depth image and RGB image acquired by kinect....6
Figure 2-3: Indoor mapping, (a) entire floor reconstruction(Du et al.) (b) top view indoor mapping(Du et al.)....7
Figure 2-4: Color histogram representation of an image (Lin et al., 2002)....8
Figure 2-5: Sift descriptor (Prince, 2012)....8
Figure 2-6 : Spin image of three oriented point (Johnson et al., 1999)....9
Figure 2-7: Parameters involved in the 3d Hough transform in object detection((Khoshelham, 2007)....10
Figure 2-8: Illustration of the partial matching process (Kanezaki et al., 2010)....11
Figure 3-1: Framework of the methodology....13
Figure 3-2: Filter and convolution....16
Figure 3-3: Example category of object, from left to right: dry battery, flash light, bowl, and keyboard....17
Figure 3-4: Illustration of bow model in document representation("Text Classification in Python,") 18
Figure 3-5: Illustration of the bag of words workflow....19
Figure 3-6: Simple illistration of the principle of svm (a) data sepsrate in two classes (b)separate data by mapping it into high dimension feature space....21
Figure 3-7: Illustration of principle of svm("The Standard SVM Formulation,")....22
Figure 3-8: Image of real scene....23
Figure 3-9: Classifier based illustration....23
Figure 4-1: Multiple view of the cap....25
Figure 4-2: Images of big trainning dataset....26
Figure 4-3 : Images of small trainig dataset....26
Figure 4-4: (a)Original RGB image of cap(b) normalized grayscal image....27
Figure 4-5: Gradient image,colour bar indicates the edge change rate, the larger value in the colour bar, the larger the image change over gradient (a) gradient image over horizontal direction (b) gradient image over vertical direction (c)overall gradient magnitude image....28
Figure 4-6: Grid point distribution and patch level gradient image....29
Figure 4-7: Depth image after normalization....29
Figure 4-8: Depth image and gradient image....30
Figure 4-9: Patch-level gradient image over depth image....30
Figure 4-10 Point clouds of the objects....31
Figure 4-11: Coordinate system of the depth image of Kinect. circle is the object, Triangle refer to the kinect....31
Figure 4-12: Illustration of removing of the incorrect point .black points are irrelevant points and red points are chosen point....32
Figure 4-13 (a) Random sample points, red dots in the point clouds (b) where black triangular is the selecting key point for one patch....32
Figure 4-14: Histogram representation of the distant measurement over different objects. The x axis denote the number of the point cluster, and the value over y axis is the distance measurement with unit meter. ....33
Figure 4-15: Generation of bag of words....34
Figure 4-16: Illustration of the bag of words....35
Figure 4-17: Histogram representation of RGB gradient features over different objects, each histogram only represents one training image of an object....36
Figure 4-18: Histogram representation of depth gradients features over different objects....37

Figure 4-19: Sample number with respective accuracy ..... 38
Figure 4-20: Number of visual words with respective accuracy ..... 39
Figure 4-21: Training flow chart ..... 39
Figure 4-22: Test RGB image and corresponding depth image ..... 40
Figure 4-23: Partitioned windows from the test image ..... 40
Figure 4-24: Windows classified as the cap ..... 42
Figure 4-25: Visualisation image of the recognition result,(a)ground truth bounding box (b)initial windows classified as cap (c)mean bounding box over the windows previously. (d) Final recognition result. ..... 42
Figure 4-26: Recognition result using size descriptor (a)original bounding box recognized as cap (b)final result ..... 43
Figure 4-27: Visualisation image of the recognition result over the combination of RGB image and depth image ..... 43
Figure 4-28: Multiple object recognition result the red bounding box refer to the cap, the blue one denote the flashlight (a) recognition using RGB descriptor (b) recognition using the combination of RGB and depth features ..... 43
Figure 4-29: Coffee mug recognition result ..... 44
Figure 5-1: Overall accuracy on different descriptors ..... 46
Figure 5-2: Overall accuracy over the small dataset ..... 46
Figure 5-3: Illustration of ground truth box and window being recognized (a) the black bounding box denote the ground truth box(b)red box is the window classified as cap ..... 48
Figure 5-4: Point clouds of the image of real scene ..... 50

# LIST OF TABLES

Table 4-1: Gaussian filter ..... 27
Table 4-2: gradient filter over horizontal and vertical direction, value being normalized ..... 28
Table 4-3: local features and corresponding size of the vector, specifically, N,M,P denote the number of the patches per image ..... 34
Table 4-4: sample number with respective accuracy ..... 38
Table 4-5: number of visual words with respective accuracy ..... 38
Table 4-6 Dataset of posterior probability ..... 41
Table 5-1: overall accuracy on different descriptors over big dataset using qdc ..... 45
Table 5-2: overall accuracy on different descriptors over small dataset using svm ..... 46
Table 5-3 : confusion matrix of the overall accuracy ..... 47
Table 5-4: cap recognition overall accuracy over different descriptor ..... 49

# 1. INTRODUCTION

## 1.1. Motivation and problem statment

Indoor modelling is the process that generates digital or graphic representations of the physical and functional characteristics of indoor environment. Nowadays, adequate and accurate indoor information is playing a more and more important role in many fields, such as interior design, robot navigation and surveillance (Thrun, 1998). Traditional indoor modelling is provided by blueprints made by CAD. However, because a blueprint is drawn before the construction of the real building, it lacks the current building information as well as concrete interior objects information.

3D modelling of building interiors from actual data acquisition solves the problem above. Modelling from walls and floors to object and other indoor objects by using camera or other sensors can provide broad location information about the environment as well as rich semantic information. For example, establishing a system of virtual 3d indoor model with real time acquisition can help people interact with the virtual reality of the interior environment (Hao et al., 2011).

In order to build the indoor models, the first procedure is to recognize walls, floors, ceilings and objects of indoor environments in the acquired data. The present research will only focus on indoor objects, such as furniture, vase and other household objects. The purpose of the object recognition is to determine the identity of an object being observed in the data. Visual inspection by human being is very easy while it is not the same in the case of computer recognition.

Object recognition from 2D images is an appealing approach due to the widespread availability of cameras. However, 2D recognition techniques are sensitive to illumination and shadows. 3d object recognition, on the other hand, does not suffer from these limitations (Mian et al., 2006).

Different approaches have been developed for 3d object recognition. Image-based 3d object recognition, which presents a robust recognition of outdoor scenes, has achieved little success in indoor object recognition due to the limitation of low lighting of the indoor environment. Using the point clouds acquired by laser scanners carried on tripod or robot to recognize the indoor objects (Frome et al., 2004) is not only costly but also inconvenient, which is not feasible for the ordinary consumer.

As an emerging and potential technology, Kinect (Microsoft) shows great strengths over other sensors in object recognition. Kinect is a low cost structured light camera, which is easy and convenient to carry. By capturing RGB (visual) images along with per-pixel depth information, an RGB-D dataset is generated. In practice, we have a point cloud with colour information per point. It is possible to build 3D point clouds using depth information, which is well appropriate for 3D reconstruction and frame-to-frame alignment. Such RGB-D data are very suitable for the recognition of 3d objects in indoor environments.

As mentioned above, the image based 3d object recognition method has the advantage of having rich colour and texture information but lack accurate geometric and positional information(Gevers et al., 1997). Whereas methods only based on depth information show accurate geometric information of the objects but are limited in acquiring information on less texture surfaces. Our research aims at taking advantage of

both methods, to develop an effective and robust methodology which combines the colour (RGB) and depth (D) data using the RGB-D data acquired by Kinect to recognize indoor objects.

## 1.2. Research identification

### 1.2.1. Research objective

The overall aim of the research is to develop a methodology to recognize one or several specific objects in RGB-D data of an indoor environment. Two types of recognition tasks are expected to achieve:

1. Self test recognition.
2. Recognition of data of real scene which contain multiple objects and background.

It should be noted that, in this research we are majority focus on the second task and propose or improve relevant methods. The result of the first one is used as an evaluation reference, so that we can modify our algorithm and improve the recognition performance.

Sub-objectives:

1) Feature extraction:

Given sufficient number of training images, our first aim is to propose or improve some feature descriptors to detect and extract meaningful representations (features) from RGB and depth data of the object.

2) Training and Classification:

In this phrase, given the features extracted in the previous step, our task is to find a method to classify the features with corresponding class labels over measured testing data using the features stored in the training data.

3) Visualization of the recognition result.

Find a suitable method to visualize recognition results over the second recognition task.

4) Quality assessment over the result.

The quality of training the classifier and further recognition result should be verified so that the developed algorithm and programme can be improved and modified.

### 1.2.2. Research questions

1) Which local features or representations of the object are important and efficient for our recognition work?
2) How to detect and extract the features from the measured data in an efficient way?
3) How to aggregate these local features extracted in the previous step in a global level?
4) How to combine the colour (RGB) features and depth (D) features in the feature extraction?
5) How to match the features extracted in the previous step with the corresponding features in the existing training dataset in an efficient way?
6) How to keep recognition work invariant to rotation, shift or change of scale?
7) How to assess the quality of our work?
8) What is the advantage of RGB-D data in comparison with colour-less point clouds or images?

### 1.2.3. Innovation

Previous work on the recognition of indoor objects is based on either images or point clouds. A few works has been done to combine these two channels to recognize the indoor objects. In my research, one major innovation is aimed at developing a novel methodology for recognizing indoor objects in RGB-D data. Including:

A complete automatic approach and procedures on object recognition over the RGB-D data.

Develop a method that combines the features over RGB, depth, and point clouds.

## 1.3. Thesis sturcture

This thesis is organised in six chapters.
The first chapter is the overall introduction of the whole thesis, including relevant background introduction, motivation of the research, problem statement and research identification.

The second chapter mainly reviews the literature, method and research that are relevant to our work. The principle of the Kinect and character of the RGB-D data is elaborated firstly, and then the research and application on indoor perception, mapping, and modelling using RGB-D data are discussed. Later, manners and techniques that have been done over the object recognition and further matching are described briefly.

The third chapter is the research methodology. It explains and describes the proposed method in detail. The general flowchart is given firstly. Explanation of each step is followed later.

The fourth chapter is the experimental results, including the data processing and visualization over that. Parameter and threshold values are also given experimental support.

The fifth chapter demonstrates the quality assessment of the training result and recognition result. Discussion and conclusion are further given.

The last chapter is the conclusion and recommendation of this work.

# 2. LITERATURE REVIEW

## 2.1. Characterization of the Microsoft Kinect

Kinect(Microsoft), an essential part of this Microsoft Xbox interactive controlling system for specific motion game, has become a brand new motion sensor device recently. Now, because of its low price and special sensor for measuring depth, kincet has attracted widespread attention in computer vision and machine learning fields for research and analysis. There are many works on object recognition, indoor mapping and indoor modelling using the data obtained by Kinect.

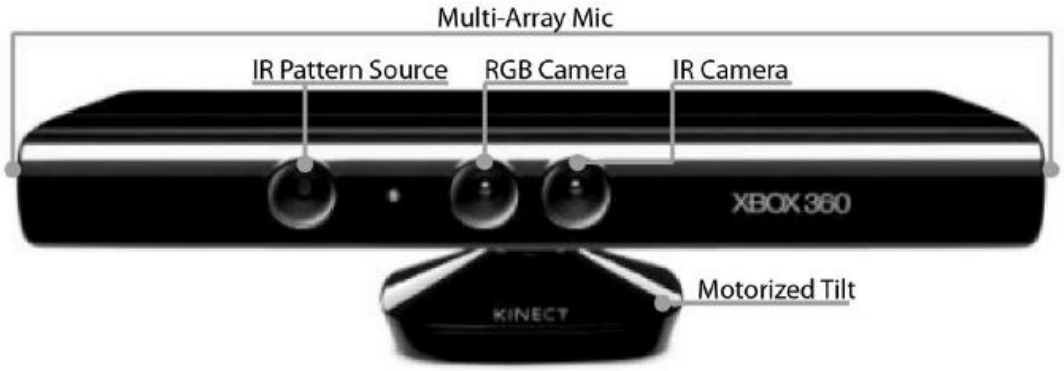
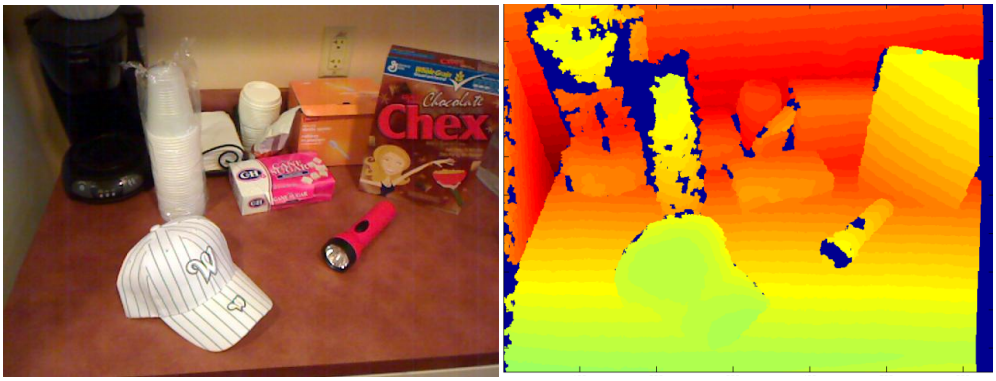



Figure 2-1:Kinect(Microsoft)

With a normal RGB camera and an IR camera (Figure 2-1), kinect is able to capture the RGB images (Figure2-2 (a)) of 640x480 pixels in 8 bit depth with a Bayer colour filter together with a same size depth image. The way to capture depth image is light coding developed by the company PrimeSense (PrimeSense). Using an IR pattern Source to project a complicated pattern of dots onto the object, the IR camera then receives the pattern image which has been illumined on the object.

The depth image is 2 dimensions, and contains information over the distance from sensor with the surfaces of the object or the scene. Limited by the hardware, kinect can only measure the object with a range of 0.7–6 m (openkinect). Meanwhile, depending on the reflective character of different objects, the depth camera would sometimes fail to capture data. Illustrated in the following Figure 2-2(b), the blue area in the depth image are missing depth values. This characteristic has aroused problem for further application.




(a) RGB image (b) Depth image

Figure 2-2: Depth image and RGB image acquired by Kinect (Kevin et al., 2011)

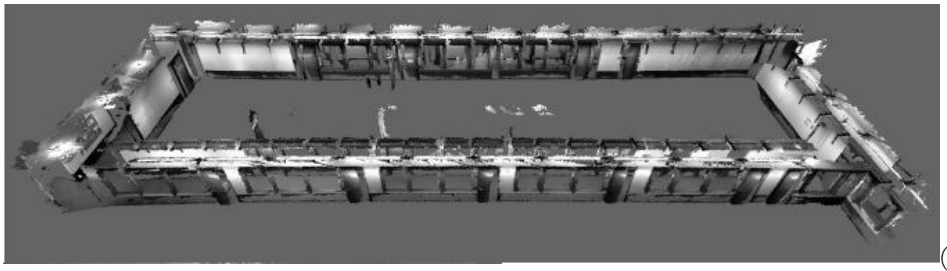
## 2.2. Indoor mapping and modeling using RGB-D data

With the wide availability and emerging trend of RGB-D cameras, the application of RGB-D camera in terms of indoor perception and understanding is increasing rapidly. A lot of research has been done using RGB-D camera for 3d indoor mapping and modeling, scene and object recognition, robot manipulation and other significant tasks.

3d indoor mapping aims at efficiently and quickly modeling the indoor environments and giving relative accurate color and location information of the scene and indoor objects. Henry et al. (2012) proposed a method to construct several frames together using an optimized algorithm that can detect loop closure and optimize the global pose. By using RGB-D ICP alignments frame to frame, it robustly and accurately built the indoor maps, which also proved a better performance than either image or geometry based alignment mapping.

In another research, Du et al. (2011) constructed a interactive system that allows the normal user to move the RGB-D camera and scan the interior space to build the 3d indoor models simultaneously. Colour and depth information were well exploited and registered together to build the 3d model. It had used a RANSAC matching between two frames to remove the outliers and locate the transform between them. This method had achieved great result that allow the user to interactively detect the failures and assist to reach complete scene coverage as well as demonstrate a promising application based on this constructed indoor model, see Figure 2-3.

Collet et al. (2011) developed an approach to generate a scene segmentation with a ranking mechanism to separate a scene into a meaningful object, while discarding background clutter. The strength of it is the use of a novel region based image and range data fusion technique to combine the image and range data for further discovery. Moreover, without a rigid assumption about the scene structure, it had obtained a better generality in the discovery of indoor scenes.


(a)

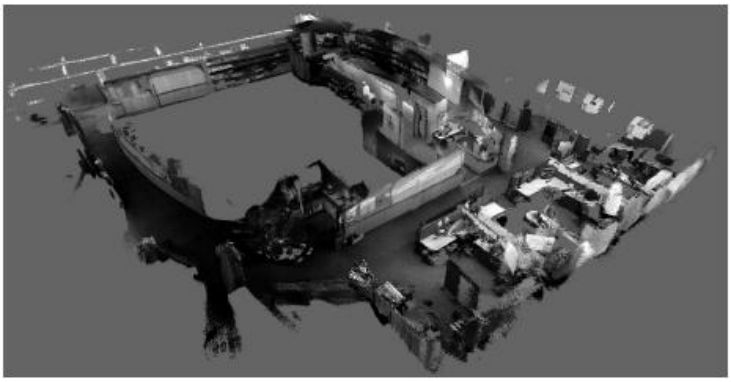

(b)

Figure 2-3: Indoor mapping, (a) entire floor reconstruction(Du et al.) (b) top view indoor mapping(Du et al.)

## 2.3. Obeject recognition by using Feature descriptors

A remarkable survey of object recognition can be found in (Campbell et al., 2001). A common approach to object recognition is by extracting and analysing feature descriptors.

Features are information originated from the image. The first step in the recognition is to find representative features from the image. Feature descriptors are algorithms that summarize the contents or the features of an image region. In addition, the extracted features are stored in vectors known as feature vectors or descriptors (Canny, 1986). In the past years, numerous methods on feature descriptors have been developed. According to Tangelder et al. (2004), the approaches on feature descriptors can be classified as histogram-based, transform based and 2d view- based.

In the following section, we will review some feature descriptors which are highly related with our research.

### 2.3.1. Histogram-based descriptors

Histograms provide a way to get together the information over a large region and calculate a distribution of the representation in this region, see Figure2-4. For instance, collecting gradient , edge cues and local binary patterns over depth image and storing them into histograms according to their character and quantity.

The histograms items can be existed as discrete or continuous, the former one can be modelled as a distribution of category, while the latter one can be quantized as vectors. Classifying the representations into many bins is a fundamental task for further discrimination. It should be noted that the quantization is rigorous for those continuous quantities. However, if the extracted features are too sparse, a lot of bins in the histogram would be empty. As a consequence, the further classification or recognition would not be credible (Prince, 2012).

This approach has a limitation in characterizing structured objects, which means spatial variance of the object would influence the identification.

In the work by Hetzel et al. (2001), they developed a method using a multidimensional histogram to store and quantize the extracted features. The multiple features are combined in this multidimensional histogram so that a comprehensive description is extracted. Later the comparison between the model objects and images from real world scene is done by matching the histogram. It has shown a robust recognition result with resistance to occlusion.

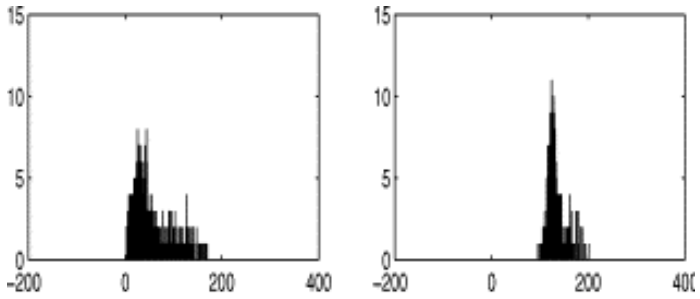



Figure 2-4: Color histogram representation of an image (Lin et al., 2002)

The Scale-invariant feature transform (or SIFT), published by David Lowe in 1999 (Lowe, 1999), is a very classic algorithm applied in computer vision to detect and describe local features in images. It characterizes the image region around a given point in the image.

At first step, a set of distinctive key points in the image are detected by certain means (e.g. DoG operator). Then by defining a region around each key point, the region content is computed and transformed into local region value (Prince, 2012). For instance, take a 16x16 square window around this detected region, whereby, a 4x4 grid of cells would be generated on the basis of this window. Further on, within one individual cell, an 8D histogram of orientation is calculated, see Figure 2-5. The value of histogram is weighted by gradient amplitude and distance. Collectively, we get a 16cells*8 orientation=128 dimensional descriptor. For further convenience, the histogram values are normalized to unit length to reduce effect of illumination change.

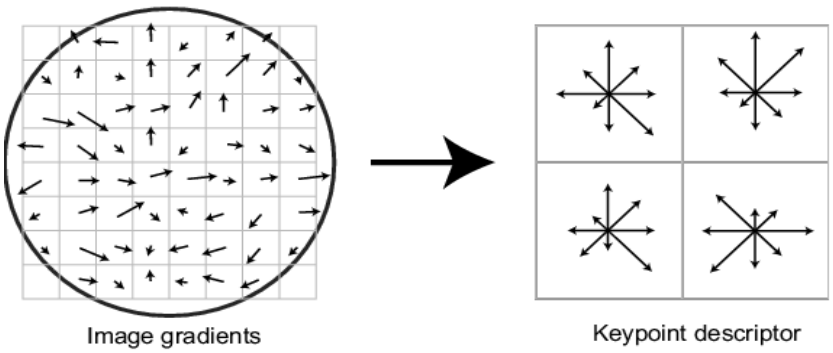



Figure 2-5: Sift descriptor (Prince, 2012)

This method is an extraordinarily robust matching technique. It is invariant to intensity and contrast changes because it is based on gradients, Also as it pools the information within each cell, it can withstand some geometric deformations. However, the drawback of this approach is that it is a little hard to design and difficult to include information other than gradients.

Specifically, another disadvantage of this method on object recognition work is discussed below: with the introduction of the vast number of key points, some irrelevant points or background of the image might provide a negative impact on the later feature extraction of the specific objects. In the work by Pavel et al. (2009) , they developed an approach using a matching tree to remove the unimportant key points to overcome this problem. In addition, one 3d model is built in the feature space after the filtering of the sift features, which is intended to enhance the ability of being invariant to rotation in the recognition.

### 2.3.2. Feature Descriptor over 3d data

With the wide availability of 3d sensors and the development of the 3d image application, more and more emerging techniques on extracting features from 3d data are applied in object recognition.

One classical feature descriptor over 3d points is spin images, which has been widely applied to 3D objects recognition problems. A representative work done by Johnson et al. (1999) is elaborated below. Firstly, densely distributed oriented point over the surface mesh of the 3d objects is produced. Then three major parameters of the other points of the objects are computed. A spin image for one oriented point is constructed by accumulating these parameters, see Figure 2-6. Further on these spin images are processed with correlation and the surface matching is established by finding point correspondence on the basis of that. The recognition between the model and test image could be obtained using this idea.

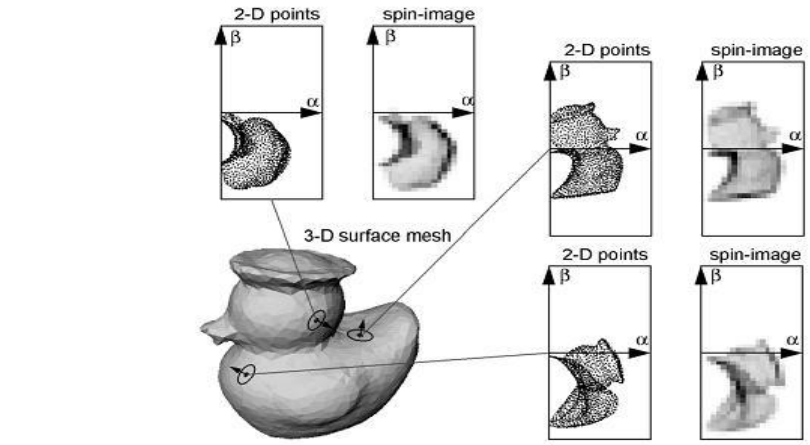



Fig. 2. Spin images of large support for three oriented points on the surface of a rubber duck model.

Figure 2-6 : Spin image of three oriented point (Johnson et al., 1999)

Another method that works on the 3d points is Fast point feature histogram (Rusu et al., 2009)) However, these methods are originally proposed for the use of 3d point clouds and are not adjusted well to the depth image acquired by kinect.

Color–CHLAC feature (Kanezaki et al., 2010) is an extension of CHALAC features (Kobayashi et al., 2004). It first transforms the 3d point clouds into dense 3d colour voxel data, then the local features is computed by the correlation function of the points over the 3d voxel data, this correlation function consider the local patterns over the 3d colour voxel data. The computed features are further compressed using principle components analysis. This work had shown an efficient and potential performance over the object recognition using 3d measurement.

Recently, a new method which uses kernel features to model size, shape, and edges of the object was firstly applied only on RGB image(Bo et al., 2010), and produced encouraging result. A development that applying the kernel principle on both depth image and RGB has been proposed by Kevin et al. (2011). It firstly adopts some classic local feature descriptors such as gradient descriptor, local binary pattern

descriptor, size feature descriptor, spin image descriptor, etc. These low dimensional local features are then transformed into a high dimensional feature space using kernel principle, so that the crowded feature vectors is separated in the high dimension space. This work outperforms those methods using traditional 3D features and greatly improves the capabilities and accuracy of depth and RGB-D recognition of object (Kevin et al., 2011)

### 2.3.3. Other approaches to object recognition

#### Bag of words concept

Methods on descriptors reviewed above mainly attempt to extract features over a small region or area around interest points. Methods on bag of words try to summarize a big region or the whole image by computing the statistics of the descriptors. An application of object recognition to robot navigation using bag of word model was developed by Jinfu et al. (2011).It adopted bag of word concept to quantize the features extracted by a key point detector. Since the final histogram of the bag of word model has utilized separate bins to quantize the local features in global level. Issus on the determination of the boundary over the bin could lead to problems in further matching or classification over on flat histograms (Rubner et al., 2000). In addition, the value of the number of the visual words (clusters) could affect the performance of recognition(Grauman et al., 2003).

#### Transform based descriptors

While as the transform-based descriptors method, a representative approach was proposed by Khoshelham (2007). It presents an extension of the generalized Hough transform to 3D data, which shows a robust effect on detection instances of an object model in laser range data. The simple principle could be illustrated in the Figure2- 7. Ulrich et al. (2003) used a modification of the generalized Hough transform with an efficient hierarchical search strategy also achieved good result on object recognition in colour image. On the basis of these, Tombari et al. (2012) proposed a novel approach that matches 3D features via feature descriptor to obtain correspondences, and then accumulates evidence of the presence of the object(s) being explored by verifying the accordance of correspondences within a 3D Hough space.

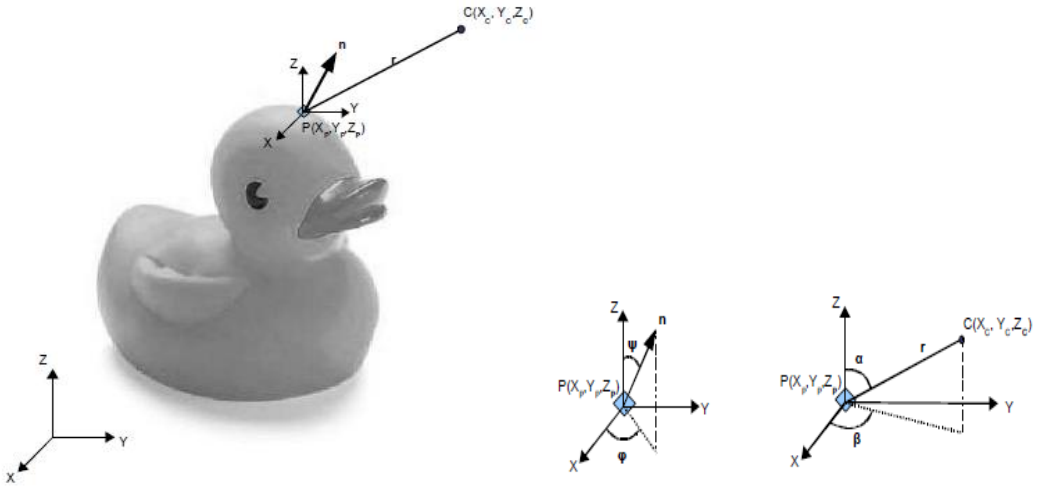



Figure 2-7: Parameters involved in the 3d Hough transform in object detection((Khoshelham, 2007)

## 2.4. Classification or matching method

After the feature extraction from the training data, a method is needed to utilize these features and find corresponding objects in a test dataset or give a decision basis for further classification or recognition of a test scene.

Two major approaches are reviewed here: matching-based and classifier-based methods.

### 2.4.1. Matching method

In terms of matching based method, obtaining correct correspondence between these features of the test image and the features of the training images is needed. Kanezaki et al. (2010) proposed a partial matching method, it first samples the 3d features by selecting a proper integration interval, and corresponding features are stored in a feature cube. Then the partial matching could be done in any part of the object. The later matching over the query region of the test object and the object in the model library is done by summing the total matching result of the sub regions. An illustration could be seen in the Figure2-8. The partial matching achieved a fast and efficient result as well as robustness to rotation. An efficient shape indexing technique was proposed by Beis et al. (1997),It matches features extracted in the test image and features from the objects model based on Euclidean distance using an improved nearest neighbour search. A new modified Kd-tree is introduced, which could find nearest neighbour in a large portion of the query objects and enhance the searching efficiency in the high dimensional feature space. Being embedded in a recognition system, this technique has shown a fast as well as an accurate recognition result.

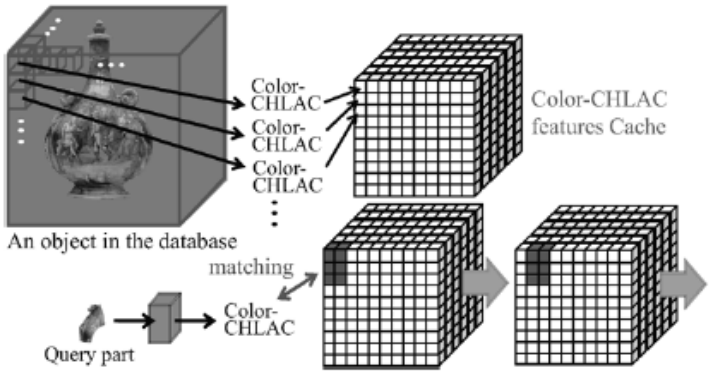



Figure 2-8: Illustration of the partial matching process (Kanezaki et al., 2010)

### 2.4.2. Classifier based method

The idea of classifier based method is that given a set of features with corresponding labels from training objects how to learn a function to predict the labels for the features of test objects.

K-nearest neighbour algorithm, proposed by Silverman et al. (1989), is an approach to classify the objects according to the closest training examples in the feature space. In short speaking, this method firstly stores all the training feature vectors together with their class label as the training set, then for each pattern or objects in the test set it searches for the closest k-neighbours based on some specific distance metric, i.e. Euclidean distance.

One research on real time object recognition proposed by Kang et al. (2007) has designed an improved K-nearest neighbour algorithm by using a modified decision rule. A local eigenspace was created to store the extracted features from the object model and serve for the matching work using newly designed K-nearest neighbour algorithm. K nearest neighbour has the advantage that it is simple and requires no training time. While on the other hand, it costs longer time for classification of the test data.

Support vector machine (SVM), a very effective and robust tool in pattern recognition field, has been proved to perform a significant function in object recognition work. The principle of the svm can be simplified as finding an optimal hyperplane that separates the features into different classes. A lot of work

using svm for object classification has been done. Pontil et al. (1998) has designed a recognition system that focuses on the generality of the svm on object recognition. It uses 7200 of 2d images containing 100 categories of objects as the training dataset. Each object is consisting of 72 multi-view images to accomplish the goal of aspect based recognition. The experiment is carried out using 36 images of one object as the training set and the remaining 36 images of the same object as the test dataset. The svm classifier is trained using training dataset. As a consequence, a file which stores the parameters of support vectors and parameters of optimal hyperplane between each object pair is obtained. Later the recognition is executed by applying the features over testing dataset to the trained svm classifier. It has obtained a relative high recognition result and indicates the well adaptability of svm classifier in the aspect based recognition.

### 2.5. Recognition over the images from real world scene

In the final step of the object recognition over images from real world scene, the goal is to identify and localize the target objects in the given image. Lai et al. (2011)used a large scale and multi-view RGB-D dataset as the training dataset and developed a sliding window manner to detect or recognize the interest objects in the testing image. This sliding window approach utilizes a score function and applies it to all the positions and scale over the test image. Next, threshold the result value of the score functions to acquire the bounding box of the target object in the test image. In addition, the background of the test image is used as the negative examples to train the window detector iteratively.

### 2.6. Summary

The RGB-D data acquired by Kinect have some advantage over the traditional sensor in indoor mapping and indoor modelling. To achieve the above targets, robust and accurate indoor object recognition is needed.

To recognize the objects, a common way is to first extract useful features from the image. Several methods could apply to RGB image, depth image and the 3d point clouds. Histogram based descriptor is a classic way to extract features over the RGB image. While method likes spin image descriptor could capture the 3d shape cues of the point clouds. A kernel based method that transforms the local features into a high dimension level combines the advantage of both channels and shows an effective recognition result. The bag of words concept could also aggregate and combine several local features into a global level. Since RGB-D data has two channels of image and features could extracted from both of them. A combination of several local feature descriptors over RGB, depth and point clouds have a potential to make a better recognition result.

To find correspondence from training image and testing image, two major approaches are reviewed. The classifier based method is more robust and more suitable to classify the test image with the labels.

# 3. RESEARCH METHODOLOGY

## 3.1. Introduction

In this chapter, the proposed methodology will be elaborated in a sequential way. Our work is mainly based on the method developed by (Bo et al., 2010) to extract the local features and then a bag of words(Jinfu et al., 2011) approach is adopted to aggregate the local features into global level.

## 3.2. Framework of the methodology

The overall proposed methodology is depicted in the Figure 3-1. Five sub-procedures compose the whole workflow, i.e. step1: feature extraction of the training data, step2; classifier training of the training samples, step3 self test recognition, step4: recognition of real scene data, step4: visualisation and result evaluation. Specifically, the step 1 and step2 could be included in the phrase of training section.

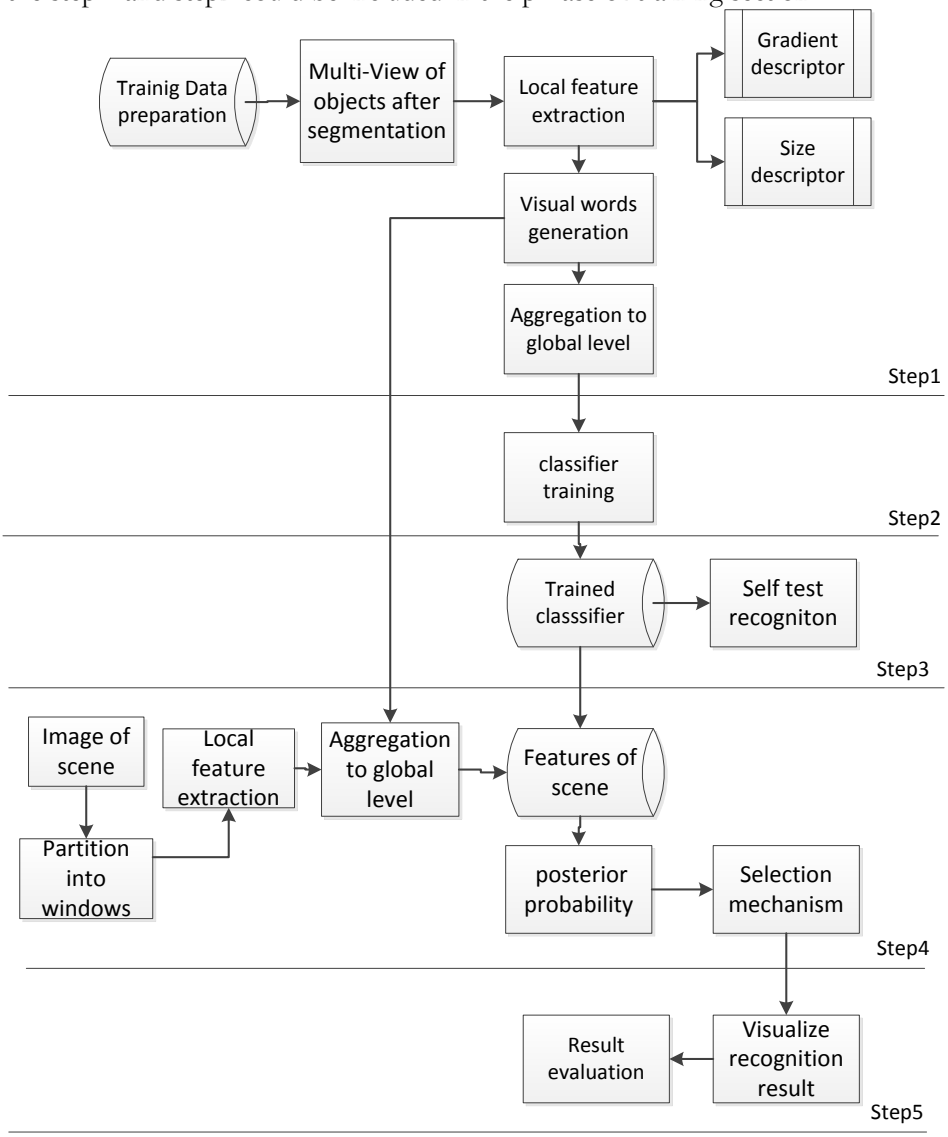



Figure 3-1: Framework of the methodology

As illustrated in the above framework, the general idea of our work is extract representative features from the training data to train the classifier, the trained classifier is then used to classify and recognize the features extracted from the test data.

Initially, original RGB-D data of the training set was processed utilizing some existing manners, and depth image is converted into point clouds for further work. Then, local features were first extracted from the training image separately. Later, a bag of words model is introduced to arrange the unordered feature vectors into numbers of visual words (clusters). Further on, those features are aggregated into a global level and one feature vector is generated to represent the whole training dataset. To be noted, the above steps only works on one individual local feature descriptors, the combination of several feature descriptors is made later in the training section.

At second phrase, we utilize two training tools with the pre-generated training features to train the classifier, which give the prior knowledge for differentiating between different classes of objects. In terms of recognition phrase, self test recognition is first conducted. We first try to assign the class label to this single object per image data sampled from part of the training data. The result of here is to evaluate the quality of the extracted features and trained classifier, meantime, provide an experimental basis to the determination of the parameters involved.

In Next step, the objective is to locate and recognize the target object in the data of real scene. The test image which contains the real scene is first partitioned into overlapping windows. The feature extraction over the partitioned windows of the test image is basically following the principle of the way over the feature extraction of the training data. Only except that, the original feature vectors over the windows are grouped and described by the histogram representation using the visual words created in the training section. After then, the trained classifier is further applied to the windows of the testing image. The posterior probability is generated after then. Based on that, a selection mechanism is developed to find the window of the target object. Finally, by locating and showing the object using a bounding box over the windows in the original test image, the recognition of the target object could be achieved.

To be noted our training data is consists of multiple instances and categories of object. Each instance is further made up of hundreds of RGB and depth image with multi-view of the object. Each image of the object is processed with segmentation to remove the background. The test data on the other hand, is images either RGB or depth that contains a full scene of the daily objects and background.

### 3.3. Local Feature extraction

Since the image either RGB, depth or point clouds are stored in their original format. To recognize specific type of objects based on the original format is very hard and time consuming. In order to better use the knowledge of the original image for further comparison. Features need to be detected and extract over the image data, Features are information originated from the image. Special method need to be operated on the RGB-D data to extract useful features for recognition. Feature descriptor is the algorithm that summarizes the image into a compressed format according to specific principle. It is widely used in the object recognition. In our designed framework, the first step is to extract important clues or representations (features) from the training images using well designed feature descriptors.

Since the daily objects we attempt to recognize is different in edge boundary and size. The edge features and size features could provide useful information in distinguishing the type of the objects.

Here we develop two local feature descriptors: gradient feature descriptor and size feature descriptor to capture the clues of the above features. Specifically, the gradient feature descriptor is applied to both the RGB image and depth image. While the size feature descriptor only works on the point clouds obtained from the depth image.

Some previous work has used interest point detector to detect and extracting features (Willems et al., 2008), (Mikolajczyk et al., 2004) , but in our work we would later use a bag of words model to yield visual words cluster based on the similarity of the features. The method of using interest point detector is not appropriate here since usually the features are too sparse for the creation of the visual words. Thus in our work, a patch based approach (Bo et al., 2011) is adopted : for both gradient and size extraction, for every image or point cloud, numbers of uniformly and densely distributed patches are used to extract local features. In our work, we initially define a certain number of grid points densely located all over the image with spacing of 8 pixels. The respective rectangular patch is created by surrounding the grid point as the centroid. The size of the patches is set as 16 pixel *16 pixel. Later on, these local features are transformed into a set of image features.

#### 3.3.1. Gradient descriptor

To better capture the edge and boundary information from the object, we utilize the gradient descriptor. Since it could capture the edge, boundary and the gradient change rate over the image in an efficient and robust way.

The image gradient measures how fast the value of colour or intensity changes along certain direction in the image. In our case, it can directly apply to the RGB image and depth image. The gradient of the image function f(x, y) at point (x, y) is a 2D vector given by the computation of derivatives in the horizontal and vertical directions. It provides two pieces of information: magnitude and direction. The gradient formula is given in below:

$$\nabla f(x,y) = [G_x, G_y]^T = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]^T \qquad 3\text{-}1$$

Gx and Gy denote the gradient along the x direction and y direction respectively. The magnitude of the gradient is:

$$\mathrm{mag}(\nabla f) = g(x,y) = \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2} \qquad 3\text{-}2$$

The gradient direction can be calculated by the formula:

$$\varphi(x,y) = \arctan|\frac{\varphi f}{\varphi y} / \frac{\varphi f}{\varphi x}| \qquad 3\text{-}3$$

At each image point, the gradient vector points to the largest possible intensity increase in certain direction, and the magnitude of the gradient vector corresponds to the rate of change in that direction. Since it tells where and how quickly the image changes in such a direct way. Thus, image gradient can be used as a very efficient tool to detect the edges or the boundary of the image.

We utilize the gradient descriptor to capture the edge cues of the RGB and depth image. The classic approach for the gradient computation is to consider the change of the gray value of the neighbourhood of every pixel. Here we use a Difference-of-Gaussian filter to compute the gradient. Because Kinect is a low resolution and low accurate sensor, the image is not very accurate. Especially the depth image contains noise and some parts of the image are blank without information. The further gradient calculation is very sensitive to noise and Gaussian filter can suppress the noise.

We first have to understand the principle of convolution and filter. Developed from the math filed,

convolution is an operation that creates a new function over two functions f and g.

$$\begin{aligned} h(x,y) &= f(x,y) * g(x,y) \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f\left(x',y'\right) g\left(x-x',y-y'\right) dx'\, dy' \end{aligned} \quad 3\text{-}4$$

If it is a discrete function, convolution function is:

$$\begin{aligned} h(i,j) &= f[i,j] * g[i,j] \\ &= \sum_{k=0}^{n-1}\sum_{l=0}^{m-1} f[k,l]g[i-k,j-l] \end{aligned} \quad 3\text{-}5$$

Here i,j can be interpreted as the pixel value in the image, n, m are the size of the filter
Applied this principle to the image processing field, convolution or filtering over the image is to replace the original pixel value by moving over the original image with specific math matrix called filter. The simplest linear filter is a partial convolution, i.e., each pixel value is replaced with the average of the values of the local neighbourhood after convolution:

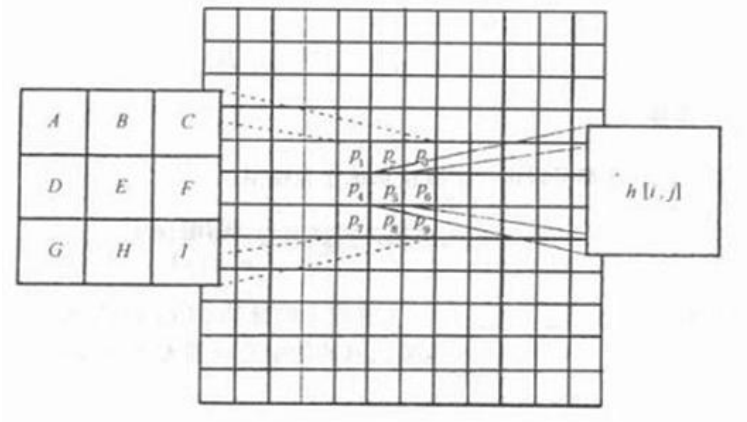



Figure 3-2: Filter and convolution(Vandevenne, 2004)

$$h[i,j] = Ap_1 + Bp_2 + Cp_3 + Dp_4 + Ep_5 + Fp_6 + Gp_7 + Hp_8 + Ip_9 \quad 3\text{-}6$$

Illustrated in the Figure3-2, the filter is a $3\times3$ sliding window that moves from the top left corner and move through all the positions over the original image. In Formula 3-4, the A, B….I denote the value of every cell in the filter, while the$p_1, p_2,\ldots p_9$ Refer to underlying image pixel value For every pixel [i, j], the output response is calculated by multiplying together the filter weighted value and the underlying image pixel value for each of the cells in the filter, and then adding all these numbers together.

The Gaussian filter uses a Gaussian function to compute a transformation that applies to every pixel in the image. Gaussian function in one dimension is:

$$G(x) = \frac{1}{\sqrt{2\pi\,\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad 3\text{-}7$$

In two dimensions, an isotropic Gaussian has the form:

$$G(x,y) = \frac{1}{\sqrt{2\pi\,\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}. \quad 3\text{-}8$$

Where x and y are respectively the coordinates along the horizontal and vertical axis, σ denotes the standard deviation of the Gaussian distribution. Values acquired from this function can further create a convolution matrix that applies to the original image. The computed new pixel value is equal to the weighted average of that pixel's neighbourhood with weights defined by the Gaussian function. In the

computation of convolution, the original pixel that needs to filter are assigned with the largest weight, and with the increasing distance to the original pixel, the neighbouring pixels obtain smaller weights. Weights are larger at the centre of the Gaussian filter and get smaller as we move further away from the centre. This characteristic guarantees the good maintenance of the edges and boundaries as well as smoothes the image and reduces the noise, which would enhance the further edge detection work.

After the creation of the Gaussian filter, we further computed the differences in vertical and horizontal direction over the Gaussian filter, thus two Gaussian gradient filters was made. These two filters are then applied to the RGB image and depth image to obtain the edge change information.

To be noted, because the RGB colour image has 3 channels and the gradient computation over these three channels are similar. We firstly convert the RGB image into gray-scale and compute the gradients of the gray-scale image. In addition, the pixel value of depth image is also normalized to [0, 1].

### 3.3.2. Size descriptor

Since the depth image provide useful information on distant measurement between objects and the sensor camera, the exploitation of the depth image would assist our recognition work. The idea of size descriptor is to attempt to catch the overall physical size of the objects, which provide a very effective way to capture the clue of different type of objects. Illustrated in Figure 3-3, Different object categories are characterized with different sizes. To be noted, this clue has a better effect in category recognition than that of instance recognition. Since that different instance object of the same category may have similar size which would lead to confusion for further training work.

Figure 3-3: Example category of object, from left to right: dry battery, flash light, bowl, and keyboard(Kevin et al., 2011)

To accomplish this target, we first convert the depth image into point clouds by mapping each pixel in the depth image to the corresponding 3d point. It should be noted that depth image has processed with segmentation so that it only contain the object without background, the 3d point clouds of the object is built. To discard the noisy or irrelevant points, we filter the point clouds by removing the points whose z coordinate value is less than 0. The further computation is based on the remaining points after filtering.

Following the same idea of using small patch to calculate the region features around the grid point uniformly and densely, we also made densely and uniformly distributed patches to extract the features in size feature extraction. In the point clouds level, the small patch is existed as the corresponding point cluster. For every point cluster, we define the key point as the centre point of the patch. Then numbers of the sample points are randomly selected from the points all over the entire point clouds of the object. In the following step, Euclidean distance between each sample points and the key point are calculated. On the basis of that, we can obtain the mean value over these measurements and this value could indicate the size cues in this point clusters level. Repeat this step for every point cluster over the entire point clouds, the size feature vector is generated later. It gives the magnitude of the size measurement over the object. The respective formula is defined below:

$$D_p = \frac{||p-k||_2}{n} \quad 3\text{-}9$$

Here, p refers to the random sample point. K denotes the key point of the point cloud. And n is the number of random sample point. $||p-k||_2$ is the Euclidean distance between points p and k.

### 3.4. Aggregating local features using the Bag-of-words concept

After the original local features are extracted, it should be noted that these features derived from one image are not in a consistent format or size due to the variance of the size of the training image. Yet, each local feature vector only characterized the contents of small regions over the images. Because later on we would use these features to train the classifier and give a similarity measurement between the trained images with the test image, Works on unifying the format of the features of each training image and testing data is needed. Besides, the aggregation of these local features to the global features is needed in order to have a global description over the whole image.

Here we use the bag of words (bow) to process the local features from previous step. The bag of words concept provides a way to summarize the local region features into a global image level. Besides, it could unify the format and size of the features of image.

It quantizes the local features of one image into separate clusters by counting the frequency of the occurrence of specific character and finally make a histogram representation, thus, the generated bag of word feature vector for one training sample would well capture the character over that training sample.
Inspired by the idea of counting the frequencies of words from a dictionary or text document (Salton & McGill (1983)), the original bag of words model aimed at retrieval keywords from the text document, by counting the frequency of the occurrence of the words. An illustration is shown in the Figure 3-4.

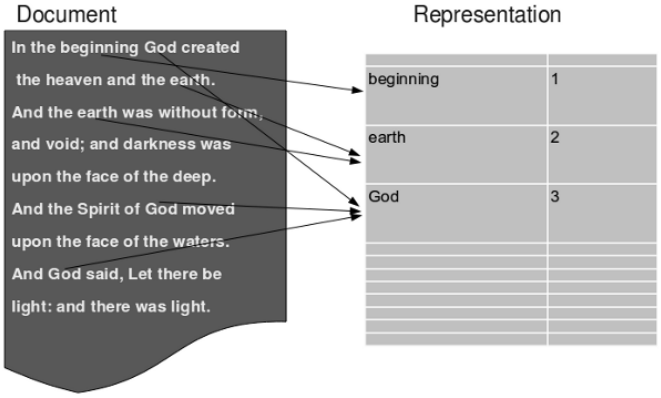



Figure 3-4: Illustration of bow model in document representation("Text Classification in Python,")

Applied in the computer vision filed, the bag of words concept provide a way to summarize a big region or the whole image by computing the statistics of the descriptor. The general procudures are elaborated below:

(1) Local feature extraction over the image. For instance in the Figure 3-5(a) the features are extracted densely form the training images using SIFT descriptor.

(2) Creation of visual cluster: using the loacl features over all the traininig images produced above, the objective here is to group the features into different visual cluster based on specific rules, for instance, by measuring the Eculidean distance. Each visual cluster is a bag of visual features which has similar characteristic. and the mean value of each visual cluster is the the representation of that visual cluster called center vector or visual words. see Figure 3-5 (e,f)

(3) Quantization of the features from one image An image could be quantizated by assigning each feature vectors to the nearest cluster center produced in the previous step.

(4) Histogram reresentation of the newimage. by computing the number of features in each visual word. a histogram representation of the image is generated, see Figure 3-5 (h,i).

A simple flowchart that illustrates the general principle and order is depicted in the Figure 3-5:

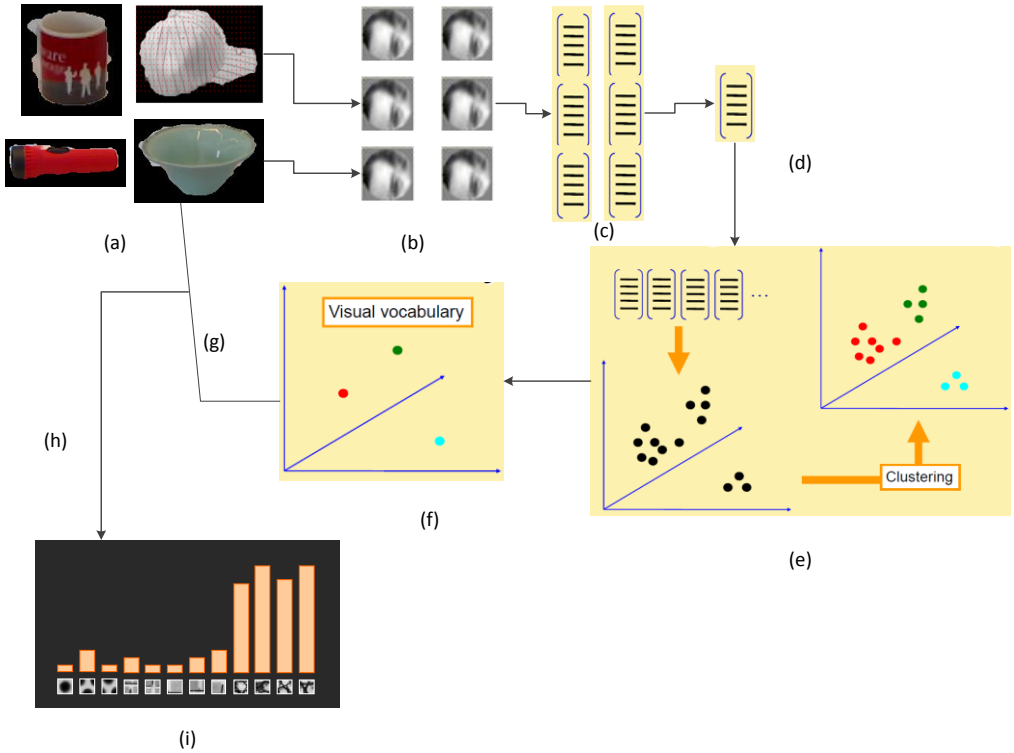



Figure 3-5: Illustration of the bag of words workflow, modified from(Hays, 2011)

(a) original image with densely distributed patch (b) extracted patch features (c) compute patch-level descriptors (d) sample step (e) sample clustering (f) creation of the centre vectors and the respective visual vocabulary (visual cluster) (g) group original feature vectors to the created cluster (h,i)histogram representation and aggregation to image level

Our proposed bag of words approach is discussed below in detail:

**1) Sampling method :**

Firstly, since we have a big training dataset and a large number of corresponding local feature vectors will be produced in previous step. To estimate the visual words (centre vectors) for each visual clusters (e, f) the number of the original feature vectors is too large. So a sampling method is adopted to save the computation time and improve the efficiency of the later clustering.

In the previous step, every training image is extracted into numbers of patch-level feature vector. We further sample n number of patch level feature vectors out of all the patch level vectors per image. Repeat this step for every feature vectors of one image and collect these sampled patch level vectors into one vector. The sampling is finished. For example, imagine that from one training image we have extracted gradients over 100 patches of size 16x16 pixel. That gives us 100 feature vectors with 256 features. For 100 training images we get 10000 feature vectors. We randomly select for instance 10 feature vectors per image resulting in a total of 1000 feature vectors with 256 features each.

**2) K-means clustering**

Next, using unsupervised learning method such as K-means, those extracted descriptors will be grouped into numbers of visual clusters (see Figure3-5(e, f)). K-means clustering is a way of cluster analysis that

attempts to divide n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

Initially, we randomly select a set of K instances as cluster centre (cluster mean) of the cluster. Then, by computing the sum of Euclidean distances between all the sample feature vectors generated in previous step and each cluster centre, each sampled feature vector is assigned to the nearest cluster centre, a cluster is generated by grouping each new assigned sample vectors. In addition, cluster centres need to be recomputed as the mean value of the new clusters. For every instance assignments, the recomputation of the cluster centre is carried out. We iterate the above step until the assignments are finished and k number of centre vectors is finally determined and the visual cluster is created (see Figure 3-5(f)). For example, imagine that we plot our 1000 feature vectors in a 256 dimensional space. Those feature vectors that contain similar characteristics of a training object would fall close to each other forming clusters. Now by doing a k-means clustering we can cluster all feature vectors into for instance 50 clusters and calculate cluster centres. These 50 cluster centres will be our visual words.

**3) Aggregation from local level to global level :**

In the above step, feature vectors over one image are presented in the format of a set of local patch features. Moreover, these feature vectors are independent as one image and did not represent the entire training images as a whole. The demand for the aggregation from local patch level to the global image level is significant.

The original patch level feature vectors could be quantized and group into different visual clusters by assigning them to the closest visual words (cluster centre) made in the previous step. Thus, for one image we could make a histogram representation by counting the number of patch-level feature descriptors assigned to each cluster. One histogram representation is the bag of words descriptor for that image. We further collect all these histograms which represent the entire training image into one vector.

Continue with the example of above, for one training image, all the 100 patch level feature are assigned to the closest centre vectors made in K-means clustering phrase. One histogram representation with 50 visual clusters is made later. Finally, if we have 200 images, the 200 histogram representations for all the images are collected into one vector.

The bag of words approach has the advantage that can deal with a variety of information about the object. It aggregates the local features into global level as well as arranges the unordered feature vectors into an ordered sequence, which makes further work more robust and efficient.

## 3.5. Classifier training phase:

The task of later recognition work can be regarded as a classification problem. The histogram representation for every image made in previous step is then used as a training sample to train a classifier so that we can later classify the test images obtained by features of test images. The global features over RGB gradients, depth gradients and size descriptor can be trained separately or combined in one joined features for training.

Here we use classifiers as the training tool: Feed with the training samples with the corresponding class labels, the classifier aim at finding a discriminant function learned from the training samples, this learned discriminant function can be further used to predict the labels from the features of unknown data. In this work we utilize two types of classifiers as described below: support vector machine (svm) and Quadratic

discriminant function classifier (qdc). The former one is used to train our smaller dataset and the latter one is used to train the big dataset.

### 3.5.1. Support vector machine

Support vector machine(svm) first introduced by Vladimir N.Vapnik and the current standard version (soft margin) was proposed by Vapnik and Corinna Cortes in 1995 (Cortes et al., 1995). The SVM is a powerful tool used for classification and regression in the pattern recognition field. The basic idea is that given numbers of training dataset together with their class label, the svm construct a mathematical model that separates the different classes' labels by defining the decision boundary.

Take a very simple example illustrated in the Figure 3-6. The dataset in the original feature space need to be divided into two classes. To separate them correctly, giving a right decision boundary is required. But for those data or input values which are merged together, the decision boundary is not easy to draw. Using a series of math functions namely kernels, SVM transform or mapping the features from the input space to a high dimension feature space (Figure 3-6(b)), after then a decision boundary called hyperplane is used to distinguish the classes. Thus the dataset or features which are not linearly separable becomes linearly separable.

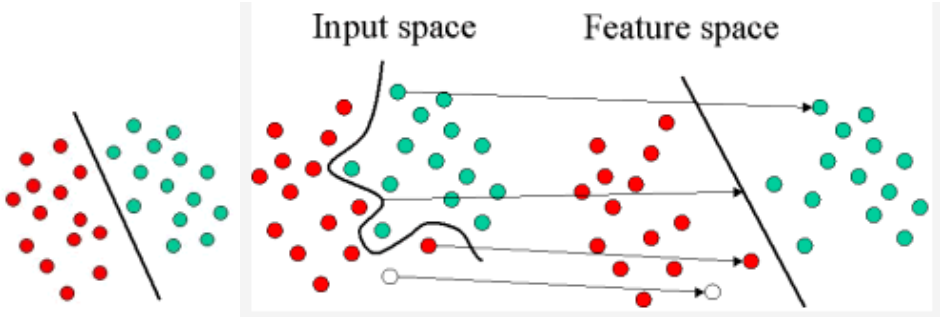



Figure 3-6: Simple illistration of the principle of svm(Scholkopf et al., 1999)

(a) data sepsrate in two classes (b)separate data by mapping it into high dimension feature space

In order to distinguish the separable classes as clear as possible with a wide gap between each class, finding an optimal hyperplane that have the largest distance between each class is the target of the efficient classification. Given a training dataset D, the xi is the training dataset, and yi = {-1, 1} is the two categories to distinguish. The defined decision boundary can be written as the equation:

$$w^T x + b = 0 \qquad 3\text{-}10$$

Where w is the normal vector of the decision hyperplane, and b/||w|| is the perpendicular distance from the line to the origin.

.A graph can be seen in the Figure 3-7.The data or example of the training set that close to the hyperlane is called support vectors. To separate the data, two hyperplane are chosen so that there is no point in between them. From the formula and description above, we have the equation over two hyperplane:

$$w^T x + b = -1 \qquad 3\text{-}11$$

$$w^T x + b = 1 \qquad 3\text{-}12$$

The area determined by them is known as the "the margin". Then the objective becomes to maximize the width of margin. According to the principle of geometry, the width of the margin is $\frac{2}{||w||}$ . Thus, for the aim of having larger width of margin, our problem then becomes minimize the ||w||.

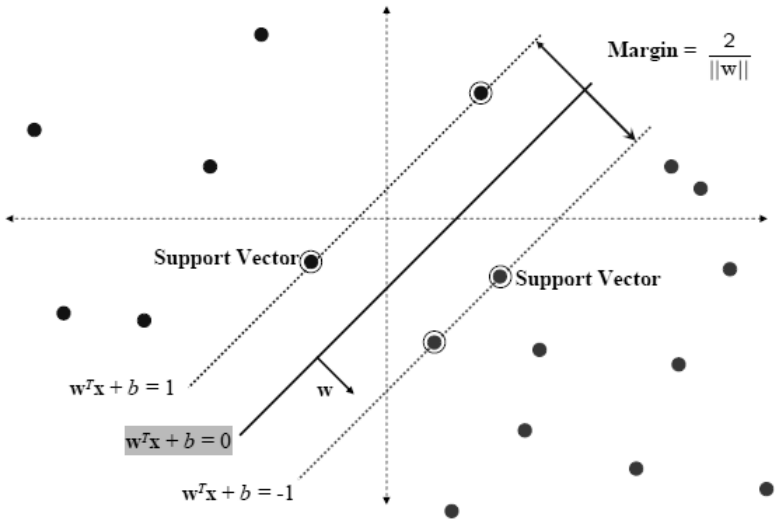



Figure 3-7: Illustration of principle of svm(Scholkopf et al., 1999)

So the problem can be transformed into a quadratic optimization problem: find w and b so that $\frac{2}{||w||}$ is maximized, for data and respective categories $\{(x_i,y_i)$, $w^T x_i + b \geq 1$ if $y_i=1$; $w^T x_i + b \leq -1$ if $y_i = -1$.

### 3.5.2. Quadratic Discriminant Function:

In terms of the Quadratic Discriminant Function classifier (qdc), it uses a normal densities based quadratic discriminant function to classify the class of data. Again we take a two class classification problem as the example. The qdc is attempting to classify the data by defining a quadratic discriminant function:

$$D_{qdc} = -\frac{1}{2}(x-\mu_1)^T \sum_1^{-1}(x-\mu_1) + \frac{1}{2}(x-\mu_2)^T \sum_2^{-1}(x-\mu_2) + c \qquad 3\text{-}13$$

Where x is a feature vector belong to the data we want to assign a class label, $\sum 1$ and $\sum 2$ denote the covariance matrices of the features, μ1 and μ2 refer to the mean vectors of the features over the two classes respectively. And c is a constant value.

The above two classifiers all have robust classification ability. The qdc is simpler and more efficient which costs a little time for training, but it does not give a very accurate indication of the soft label on posterior probability .While on the other hand, the svm requires a longer time for training but is robust in giving a significant soft label on posterior probability after applying to an unlabeled dataset. Since each classifier has its strength and weakness, we experiment with both classification methods in the later section. Specifically, the qdc is used to train the big dataset since it is faster, the svm is used to train the smaller dataset and provide posterior probability over the recognition of the images of real scene.

## 3.6. Self test recognition

In this phrase, the goal is to give prediction label of instance or category of the objects over the previously trained data. The results of here is to evaluate the quality of the features and trained classifier as well as give reference and provide experimental basis to the determination of the parameters involved.

The evaluation dataset that need to be recognized and give prediction label is sampled from our training dataset. They all have the same format: one image contains one object. We apply the trained classifier to the sampled elevation dataset, and a dataset having hard label of each class of the object could be obtained. At the same time these hard labels of each class of object are assigned to each evaluation image. The

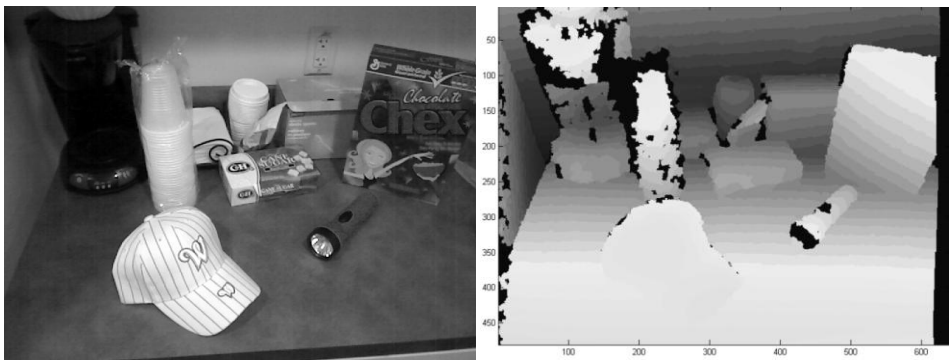
respective recognition result is then accomplished. Further quality assessment work is needed to estimate the accuracy of the prediction of recognition.

### 3.7. Recognition over images of real scene

We store the trained classifier in the system and move on to the next phase. Our recognition over the image of real scene problem can be formulated as a classification problem. The image that is needed to be recognized is an image that contains entire scene of daily objects with cluttered background. In addition, the images need to be recognized above are also displayed in RGB and depth form. An illustration could see the Figure 3-8.

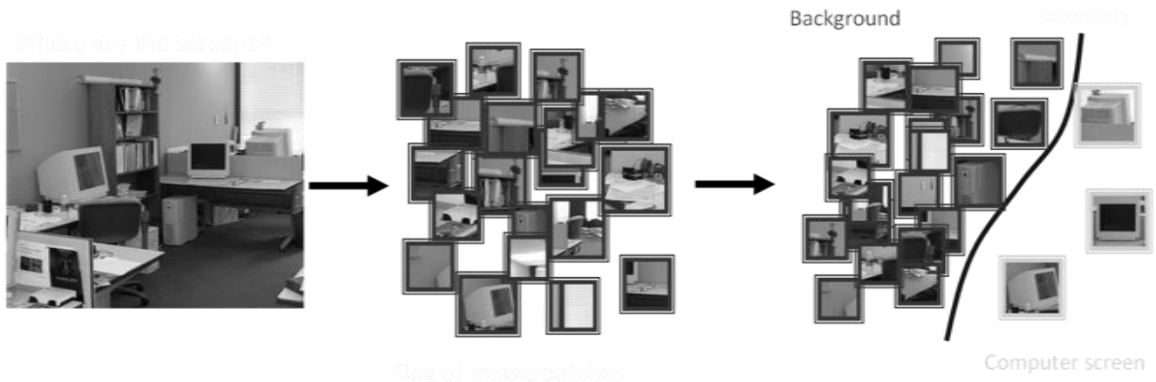
Before come into the detail of the recognition method, we first have to distinguish two types of recognition work. One is category recognition, the other is instance recognition. Our training dataset consists of numbers of categories of object, each category is further made up of several instances of object. For example, the category of cap can be made up of 4 instances of cap: white cap, red cap, black cap or black cap but in another shape. In terms of category recognition, the objective is to distinguish the cap from other category of objects, such as coffee mug. While on the other hand, the aim of the instance recognition is to distinguish each specific instance of object, for example, classify between white bowl and black bowl.

Here we partition the unlabeled image of real scene into a set of overlapping windows, and a decision is taken at each window about if it contains a target object present in the training database or not. One illustration is depicted in the Figure 3-9:



(a) RGB image (b) depth image

Figure 3-8: Image of real scene(Kevin et al., 2011)




(a) Image need to be test (b) refers to the partitioned windows(c) windows are classified

Figure 3-9: Classifier based illustration(Fergus, 2008)

Following the same principle and approach of the feature extraction step elaborated in the previous section, we extract the size descriptor and gradient descriptor over the cropped windows. Then apply these features of the windows to respective trained classifier acquired previously. We then obtain a dataset which indicates the posterior probability or confidence level of each instance or category of the object appeared in the training dataset.

The value of the posterior probability over each window is a soft label classification result of each class of the object, which indicates the probability of the presence of each type of training object. Since there are not only target objet but also the cluttered background or irrelevant objects appeared in the partitioned windows, further work need to be done to identify which window contain the target object. A selection mechanism is developed here to discard those windows containing the non-relevant objects or background in the testing image.

Here we retrieval the target object by choosing the window which has the largest posterior probability value of the target object among all the soft label result of each type of object . The windows should satisfy the following condition:

1) The value of the posterior probability over the cap class is the largest among all the value of the training objects in that window.
2) The ratio between the largest value and the second largest value in that window should beyond a threshold.

### 3.8. Visualization:

The final visualization effect of the result is necessary. We could acquire the coordinates of the vertex of each partitioned window generated in previous step and compute the mean value of the respective coordinates, one mean small window would generated later . Assign this window back to its original position in the image of real scene, we can locate and recognize the target object by showing a colour bounding box in the original testing mage.

### 3.9. Quality assessment

In this research, two quality assessment works are mainly considered:

1. Assessment of self test recognition.
2. Assessment of recognition of images of real scene

Firstly, to assess the quality of our trained classifier, we calculate the correctness rate of the prediction result. Using part of the training samples, a dataset of evaluation sample is obtained. Those evaluation sample images have the same format with the training images: one image only contains one object which also removes the background. The trained classifier is applying to the evaluation dataset, and a prediction of the instance or category of the objects over the evaluation image is obtained. Later, by comparing with the true label of the objects of the respective image, the recognition correctness rate over the trained classifier and evaluation samples is acquired. The result of this phrase is further used to as the evaluation reference over the selection and determination of the parameters involved.

On the other hand, to assess the quality of our final recognition work in the image of real scene. A pixel based classification metric is adopted. The overall accuracy of the correct recognized pixel is computed.

# 4. FEATURE EXTRACTION AND RECOGNITION RESULTS

## 4.1. Introduction

The feature extraction and later training as well as testing were all implemented in Matlab 2009a. A few Visualization effects of the point clouds were done in the point cloud mapping software (PCM) (created by EOS, ITC). Since this research is an experimental research, after specific step, the result would be assessed and evaluated. Algorithms or parameters would be modified to improve the result respectively.

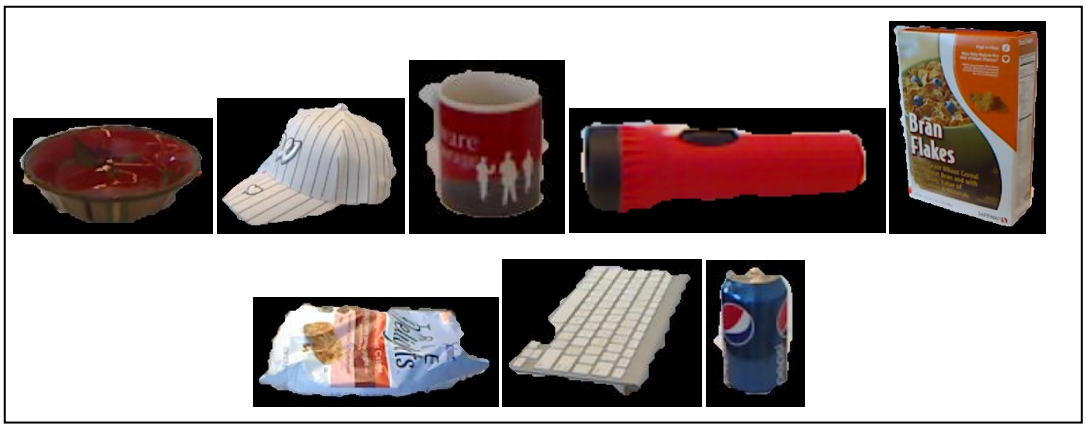
## 4.2. Training dataset

Our training dataset is collected from an open source website (Kevin et al., 2011) (http://www.cs.washington.edu/rgbd-dataset/). It is an RGB-D dataset acquired by an RGB-D camera, which contains hundreds of daily objects. For each object, the image both RGB and depth are taken in a sequence from different viewpoints. Using this rotated dataset can guarantee the further recognition work invariant to rotation, an example of the multiple view of the cap sampled from the dataset was shown in the Figure 4-1.

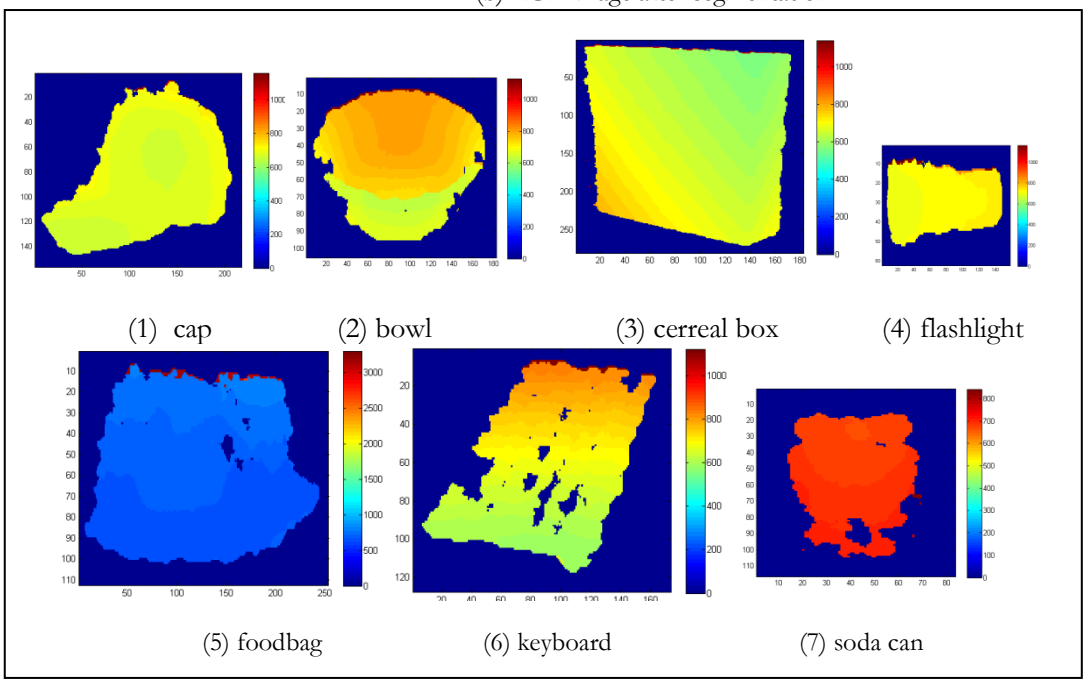
In our experiment, we first used a big dataset which contained 8 categories of objects.ie, cap, coffee mug, flashlight, cereal box, bowl, food bag, soda can and keyboard (Figure 4-2(a) (b)).the respective depth image are displayed in Figure4-2(c). To be noted, the dataset and the images are already segmented from the background. Every type of the objects consists of several instances. Thus we totally have 19 instances object for training. The number of the images of the big dataset is 2332. Meanwhile, in order to make later recognition over the images of the real world scene more efficient, we further reduced the big dataset into a smaller one which only has 751 images of 5 categories: bowl, cap, cereal box, coffee mug and flash light (see Figure 4-3).

Figure 4-1: Multiple view of the cap


(a) original training image




(b) RGB image after segmentation




(c) Depth image for training

Figure 4-2: Images of big trainning dataset

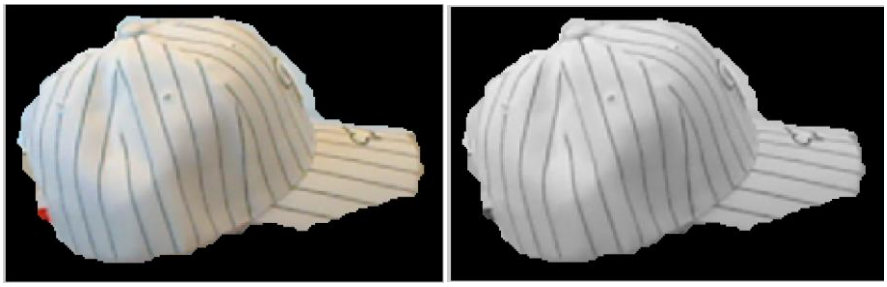

Figure 4-3 : Images of small trainig dataset

Illustrated in the Figure4-2(c), the depth images of the respective objects indicate the relative distance from the objects to the Kinect sensor. In the same depth image, the region where is farther away from the sensor has a deeper color than the region where is closer to the sensor. Another observation is that due to the character of specific material, the depth image fail to capture data in some area of the object. For example, the white area in the plastic food bag, some regions on the surface of the soda can.

## 4.3. Local feature extraction

### 4.3.1. Gradient features over RGB image

Initially the RGB image was first converted into greyscale image, then the pixel value was normalized to [0,1]. The original RGB image of the cap and the normalized image are shown in Figure 4-4:



(a) (b)

Figure 4-4: (a)Original RGB image of cap(b) normalized grayscal image

In the first step, we want to extract the gradient features over RGB image. Initially, we applied a Gaussian filter to the training dataset.

We defined several parameters of the filter: the size of the filter is 5*5, and the standard deviation set to be 0.8. The respective Gaussian filter is displayed in the Table 4-1.

Table 4-1: Gaussian filter

| | | | | |
|---|---|---|---|---|
| 0.0005 | 0.0050 | 0.0109 | 0.0050 | 0.0005 |
| 0.0050 | 0.0522 | 0.1141 | 0.0522 | 0.0050 |
| 0.0109 | 0.1141 | 0.2491 | 0.1141 | 0.0109 |
| 0.0050 | 0.0522 | 0.1141 | 0.0522 | 0.0050 |
| 0.0005 | 0.0050 | 0.0109 | 0.0050 | 0.0005 |

In order to compute the gradient over horizontal and vertical direction later, on the basis of the above filter we can produce two filters: Gx respect to x direction and Gy over y direction, which compute the differences in the two directions respectively, after then these two filters were divided by the sum of the absolute value of all the cell in the filter:

$$GX = \frac{Gx * 2}{(\sum|Gx|)} \quad 4\text{-}1$$

$$GY = \frac{Gy * 2}{(\sum|Gy|)} \quad 4\text{-}2$$

Where $|Gx|$ and $|Gy|$ are the absolute value of the cell in the filter.

The result of GX and GY is shown in the Table 4- 2

Table 4-2: gradient filter over horizontal and vertical direction, value being normalized

(a) Gradient filter over horizontal direction

| | | | | |
|---|---|---|---|---|
| 0.0102 | 0.106 | 0.2316 | 0.106 | 0.0102 |
| 0.0118 | 0.1225 | 0.2675 | 0.1225 | 0.0118 |
| 0 | 0 | 0 | 0 | 0 |
| -0.0118 | -0.1225 | -0.2675 | -0.1225 | -0.0118 |
| -0.0102 | -0.106 | -0.2316 | -0.106 | -0.0102 |

(b) Gradient filter over vertical direction

| | | | | |
|---|---|---|---|---|
| 0.0102 | 0.0118 | 0 | -0.0118 | -0.0102 |
| 0.106 | 0.1225 | 0 | -0.1225 | -0.106 |
| 0.2316 | 0.2675 | 0 | -0.2675 | -0.2316 |
| 0.106 | 0.1225 | 0 | -0.1225 | -0.106 |
| 0.0102 | 0.0118 | 0 | -0.0118 | -0.0102 |

Applied these two filters to the image, we can obtain two matrix indicate the vertical and horizontal edges of the original image. The overall gradient magnitude is the addition over the square of each direction gradient:

$$\mathrm{mag}(\Delta\mathcal{F}) = \mathrm{g}(\mathrm{x},\mathrm{y}) = \sqrt{\mathrm{GX}^2 + \mathrm{GY}^2} \qquad 4\text{-}3$$

In which GX and GY refer the gradient over x and y direction computed above. Whereby, in order to highlight the value where changes quickly and discard the not prominent gradient, so that the edge information of the object could be stressed. A tinny threshold value 1e-5 was defined. Value under this threshold of the gradient magnitude was removed.

The visualized edge image and overall gradient magnitude image are illustrated in the Figure 4-5.

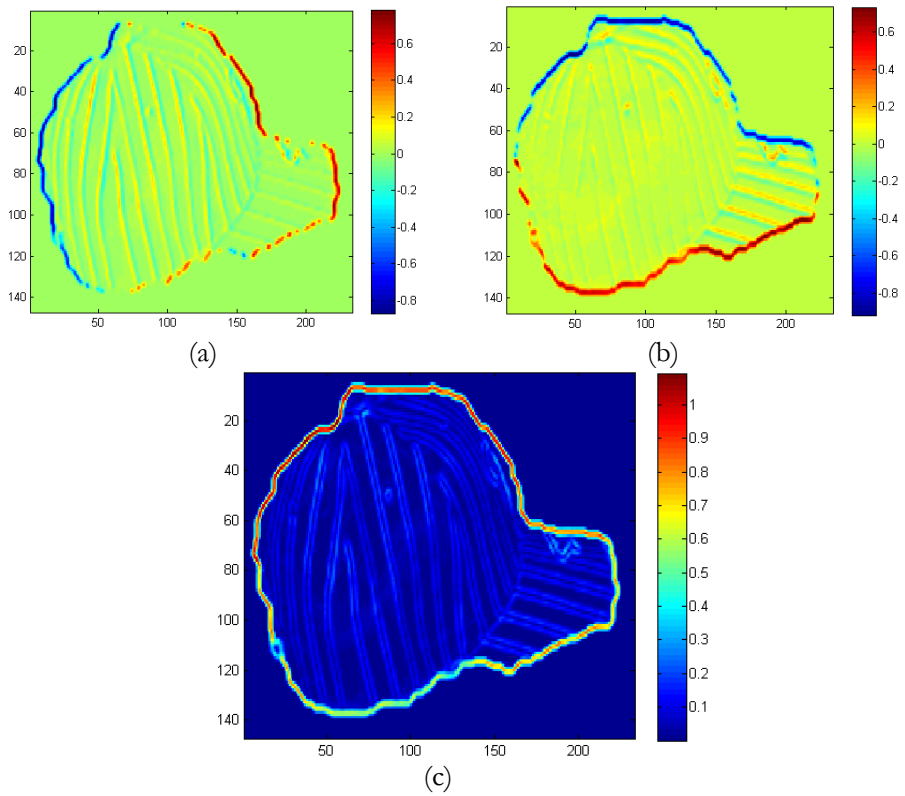
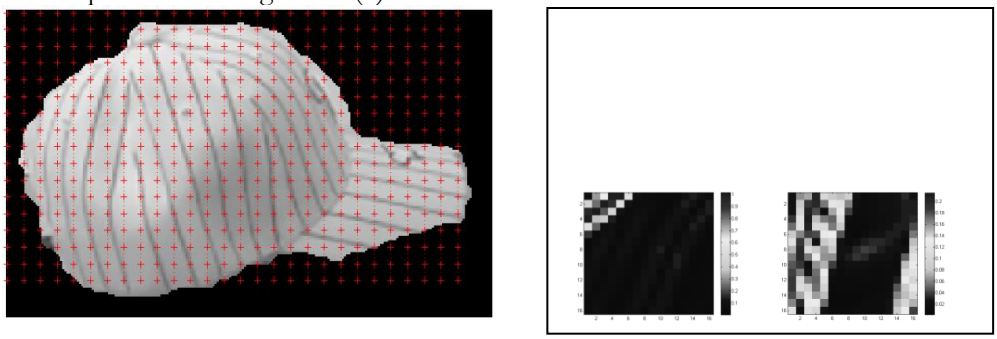



Figure 4-5: Gradient image, colour bar indicates the edge change rate, the larger value in the colour bar, the larger the image change over gradient (a) gradient image over horizontal direction (b) gradient image over vertical direction (c)overall gradient magnitude image

As can be seen in the graph, the boundary and the edges in the cap are well captured in the gradient image.

The larger value in the gradient image indicates the corresponding area has a quicker change in gradient. The gradient value over the whole image was now displayed in pixel level, we need to further transform it to the local patch level.

According to the description in the chapter 3, in the feature extraction step, we utilize a strategy to calculate the features over dense and uniformly distributed grid point with spacing of 8 pixels in the training image. Each grid point determines a rectangle shape of patch with size of 16*16 around the grid point. The grid point over the original training image is depicted in the Figure 4-6(a).
We further computed the gradient magnitude for every patch according to the size weight over the whole image. This matrix was then converted into a one column vector. A simple example of the gradient image for one patch is depicted in the Figure 4-6(b).



(a) Grid point distribution over the image (b) patch level gradient image

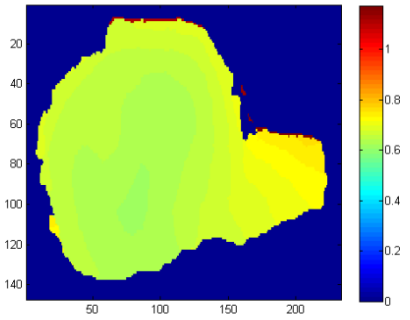
Figure 4-6: Grid point distribution and patch level gradient image

#### 4.3.2. Local gradient feature over depth image

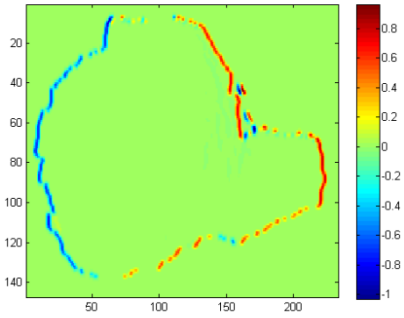
The gradient descriptor can directly apply to the depth image. The gradient filter used to compute the gradient over the image kept the same with the one used in the RGB image. Before applying the filter, some processing work for the depth data is needed. Since the original depth data is measured with the unit millimetre, we first converted the unit into meter by dividing 1000. Later, the data was normalized into [0, 1] (see Figure4-7).


Figure 4-7: Depth image after normalization

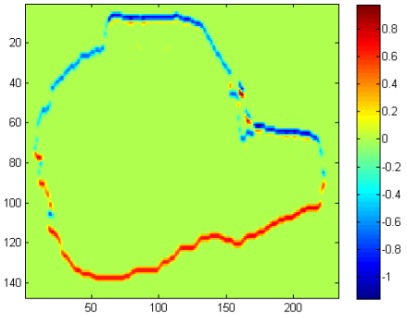
The converted depth image and respective gradient image of the cap visualised in a colour way: see Figure 4-8:



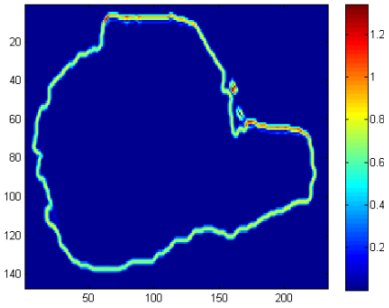
(a) Original depth image



(b) gradient image over horizontal direction



(c) Gradient image over vertical direction



(d) overall gradient image

Figure 4-8: Depth image and gradient image

Compared with the RGB gradient image, we could see that the depth gradient image mainly captured the outer boundary character rather than the internal change of the object. It can be explained by the character of the depth image, the inner intensity change is hard to be detected by the gradient descriptor. Thus, it is supposed to have a better performance in the category recognition than instance recognition.
The patch level gradient image graph, see in the Figure4-9:

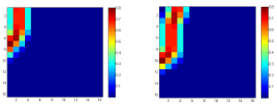
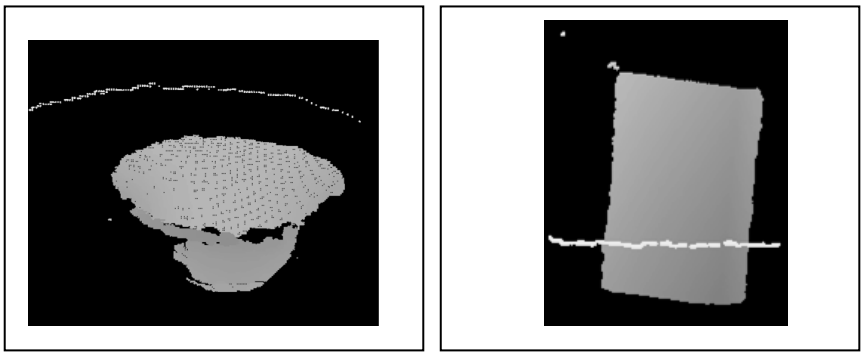


Figure 4-9: Patch-level gradient image over depth image

### 4.3.3. Local Size feature extraction:

The depth images were first converted into point clouds by mapping depth pixel value to the corresponding 3d coordinates. Some converted point clouds of the training objects can be seen in Figure 4-10.



(a) cap (b) cereal box

(c) flashlight (d) cofee mug

Figure 4-10 Point clouds of the objects

In the process of conversion from depth image to point clouds, some irrelevant points with the objects were produced. Besides some incorrect depth values were captured by the sensor in the beginning. These irrelevant points or the noisy would make a negative impact on the distance measurement over the target object, thus we need to filter out these points. The local coordinate system of the depth image of the Kinect is depicted in the Figure 4-11, the incorrect points in the point clouds has a 0 value in the z coordinate, which is not helpful in the generation of size feature over the objects. Thus, For every individual point cloud dataset, we only kept the points whose value over z coordinate is larger than 0, the respective illustration could see in the Figure 4-12.

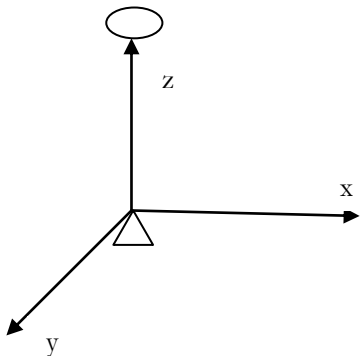



Figure 4-11: Coordinate system of the depth image of Kinect. Circle is the object, Triangle refer to the kinect

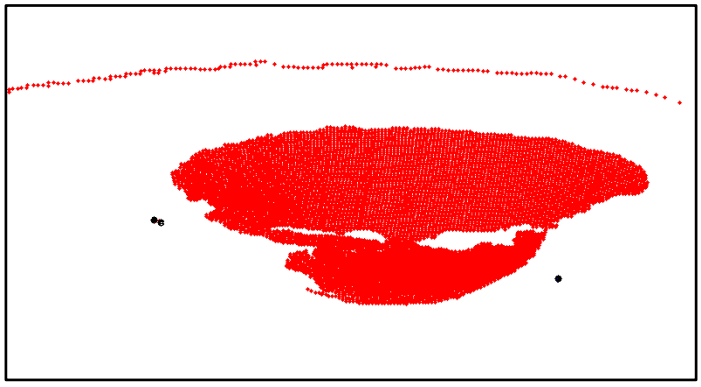
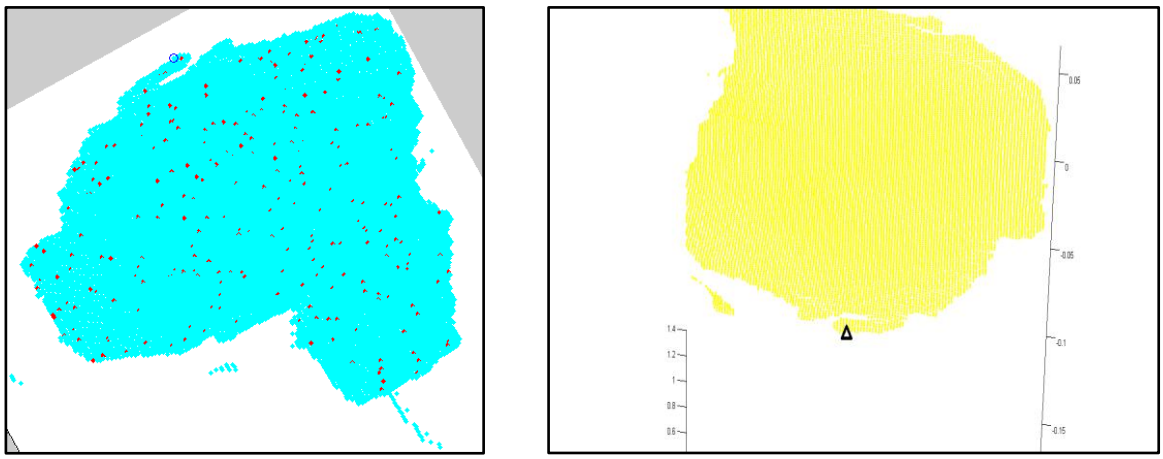


Figure 4-12: Illustration of removing of the incorrect point .black points are irrelevant points and red points are chosen point

From the Figure4-12, we could see that on the top of the objects there is still a red line of points which were not removed in the filtering step. This could lead to inaccurate computation of the size feature.
Following the same means of patch-based extraction, we had these numbers of patches with a size of 16pixel*16pixel defined by densely distributed grid point with spacing of 8 pixels. In the point cloud level, the small patch was firstly transformed into the point cluster. One point cluster was consisted of numbers of points corresponding to the pixel of the depth image. For further convince, in every patch extraction we picked the centre point of the point cluster as the key point, the centre point was the centroid computed by the mean value of the points of the respective point cluster. Then the sampled points were randomly selected from the points all over the point clouds of the objects.
An example of the key point and random sample point for one patch over cap can be seen in the Figure 4-13.



(a) (b)

Figure 4-13 (a) Random sample points, red dots in the point clouds (b) where black triangular is the selecting key point for one patch

Further on, the Euclidean distance measurement between the key point and random sample points was carried out. Based on that, we want to quantify the distance measurement of one point cluster, so the mean value over the above result was further computed. This mean value could give a basic distance measurement of the point cluster. Therefore, for every point cluster, only one distance measurement was computed and stored in the feature vector.
In order to better reveal the effect of the size descriptor on differentiating different category of objects. We made histogram representations which count the distance measurement for one point clouds over specific category of objects. (See Figure 4-14)

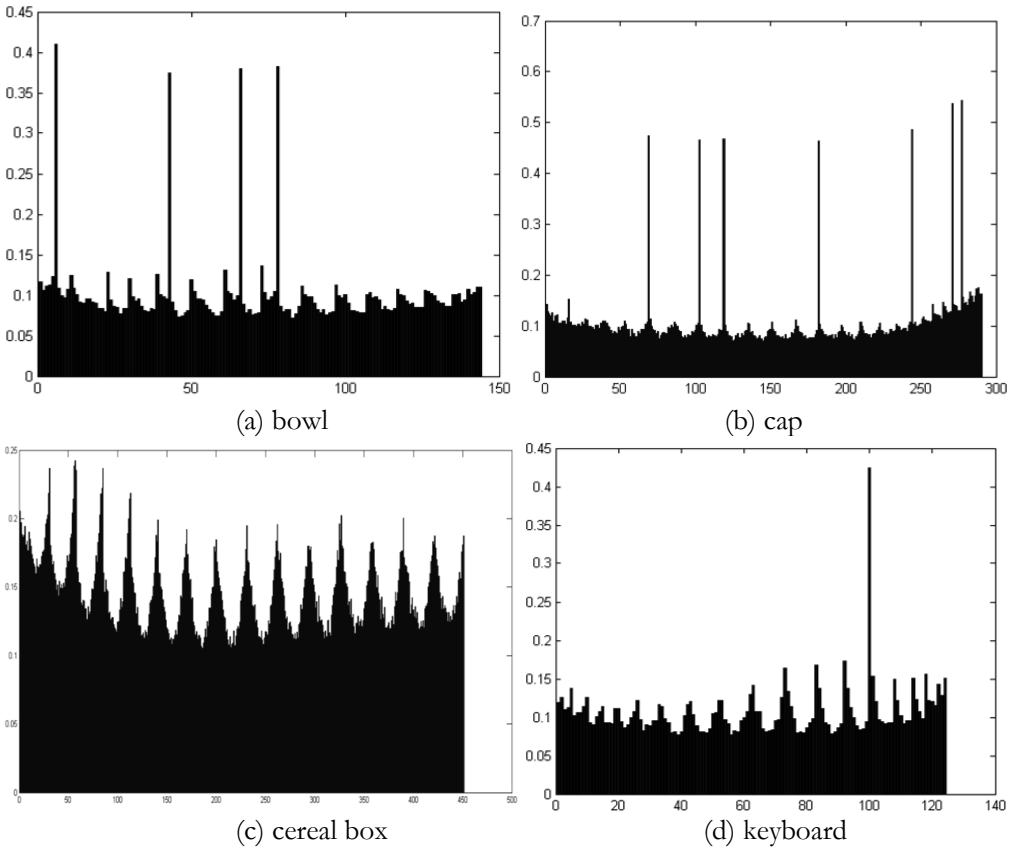



Figure 4-14: Histogram representation of the distance measurement over different objects. The x axis denotes the number of the point cluster, and the value over y axis is the distance measurement with unit meter.

As illustrated in the above graph, the magnitude of the size feature basically conforms to the size character of the objects in reality. The cereal box has the biggest size in our training dataset, and the corresponding value of local size feature is also larger than the rest of the objects in average. Besides, we could see there are slight fluctuations over the histograms. It is because in each measurement of the point clusters level, the key point was moving together with the change of the point cluster, and the sampled points were also changing each time. Another observation over the graph is that there are sudden increases of the histogram with an abnormal large value which is caused by the incorrect point clouds of the object illustrated in the Figure4-12. The abnormal value of the size would influence the further work.

## 4.4. Generation of bag of words

### 4.4.1. Centre vector and histogram representation

In the above step we had generated three local features for the training dataset. The total number of the training image is 2332 with 8 categories and 19 instances. On average there are around 100 images per instances. The name of the local features and respective size is described in the following table:

Table 4-3: local features and corresponding size of the vector, specifically, N, M, and P denote the number of the patches per image

| local features | Gradient features over RGB image | Gradient features over depth image | Size features over point clouds |
|---|---|---|---|
| number of training image | 2332 | 2332 | 2332 |
| Size of the feature vectors per image | 256*N<br>$N=\{N_1,N_2,\dots N_n,\}$ | 256*M<br>$M=\{M_1,M_2,\dots M_n,\}$ | 1*P<br>$P=\{P_1,P_2,\dots p_n,\}$ |

It can be seen in the above table that the size of the feature vectors per image varies from one image to another. It is because the image size may not be the same, which would lead to inconsistent over the number of patches per image. For gradient features over RGB and depth, they keep the same in terms of the number of the row of the vectors. The number of rows is 256 which equals to the patch size, means that for every patch-level vector contained 256 values. While, as the number of the row over the size features is 1, which means that for every patch vectors in size descriptor only one value was obtained.

We further computed the bag of words feature over each type of local feature descriptors. Initially, as the description in the previous chapter, we sampled 20 feature vectors over per image feature vector, namely, sample 20 out of N, M and P. As a result, 46640 feature vectors in total for one specific local feature descriptor was generated later. In the following step, using these sampled dataset, we created 35 number of centre vectors, each centre vector which was surrounded by numbers of sampled feature vectors is the representation of the respective visual words. The visual vocabulary is built with these classified visual words. An illustration of this process can is depicted in the Figure 4-15.

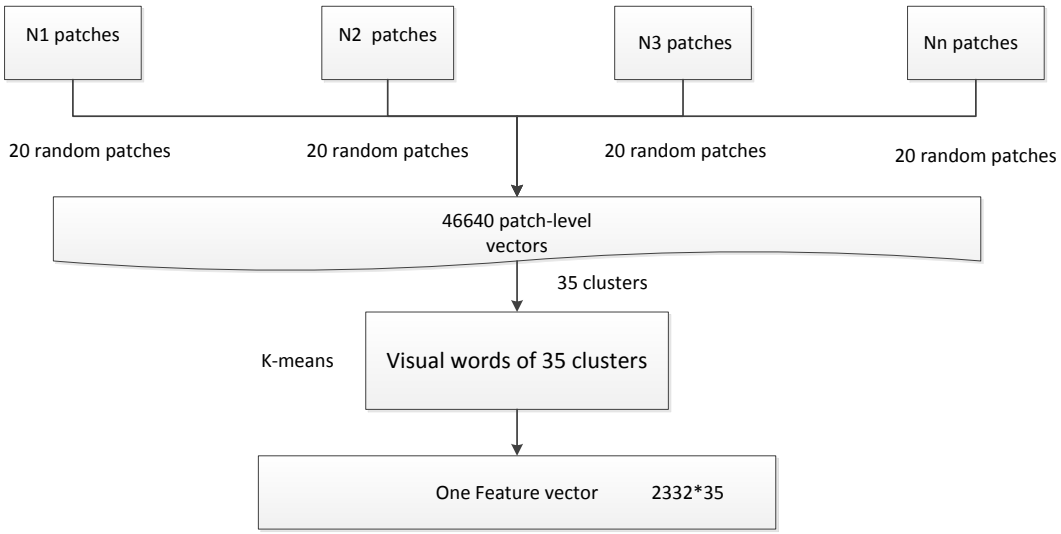
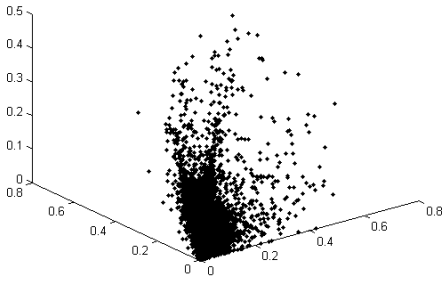


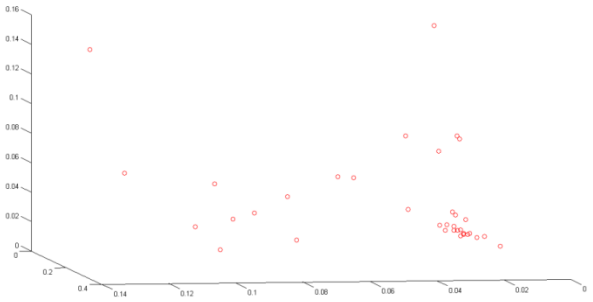

Figure 4-15: Generation of bag of words

As illustrated in the Table 4-3, the original gradient feature vector has 256 dimensions, the later centre vectors for each visual cluster kept the same dimensions with that. Here we take the generation process of bag of words over the RGB gradient descriptor as the example to show the result. In order to better visualize the distribution of the centre vectors and the created visual clusters, we picked three dimensions from these 256 dimensions( variables) ,and show the distribution of the centre vectors in the Figure 4-16(b) . The sampled feature vectors are also depicted in the same way, see in Figure 4-16(a). Since the

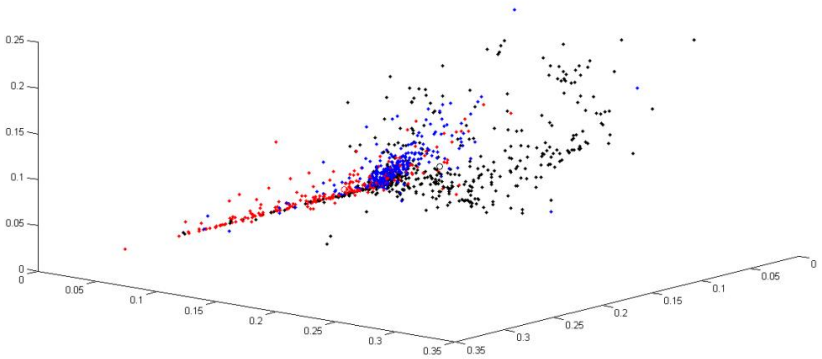
number of the sampled feature vectors is too larger, to see how they clustered in this 3 dimensions after the creation of the visual clusters is a little hard, here we only picked three visual clusters in the same 3 dimensions and show in the Figure 4-16(c).



(a)Sampled feature vectors



(b) centre vectors (created visual words)



(c) 3 visual clusters visualization in 3 dimensions

Figure 4-16: Illustration of the bag of words

From the graph above, we could see that the generated centre vectors were distributed separately in the first three dimensions. Thus the goal of separating the feature vectors has been achieved.

### 4.4.2. Aggregation and histogram representation

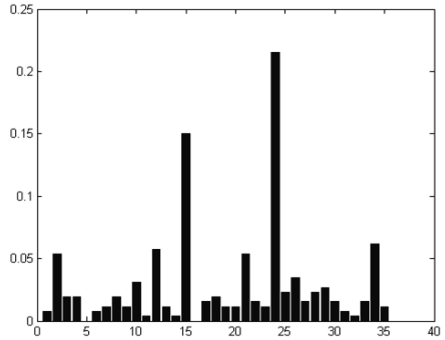
Since we had learned and created the visual words which are the centre vectors for specific visual cluster, in the next step, we need to aggregate the original local feature vectors using the created visual words.

We first computed the Euclidean distance between the original patch level feature vectors of one image with the created visual words. By minimize the distance acquired above, the patch-level feature vectors which were closest to the specific visual words were grouped together in the same cluster. Meanwhile, the number of the patch-level feature vectors in each cluster could be also obtained. Later on, the histogram distribution for every image could be computed by the following formula:

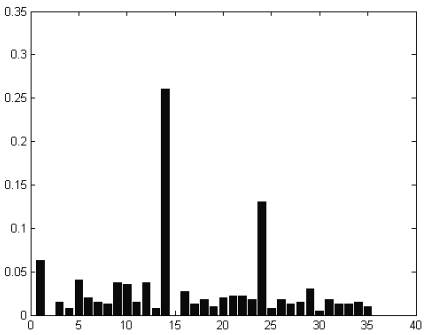
$$Hist = (n/s) \qquad 4\text{-}4$$

In which, n is the number of patch level feature vectors that each cluster has received, s refers to the size of the feature vectors for individual image. Here it is the number of patches per image, since the size of the training image varies one with another. This manner guaranteed the further classification or comparison with the testing image is in the same level. Here, we took an example of 35 visual words and

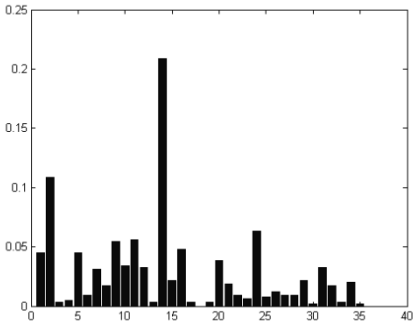
made the histogram representation of different instance of objects using the RGB gradients descriptor. The respective graphs can be seen in the Figure 4-17.



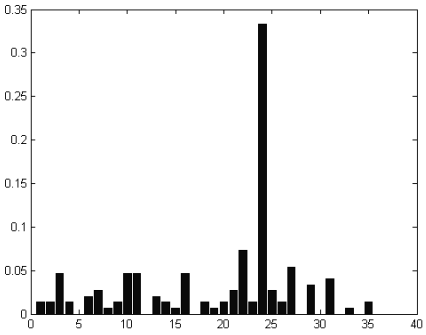
(a) bowl



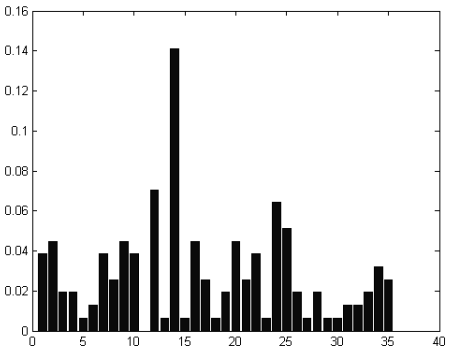
(b) cap



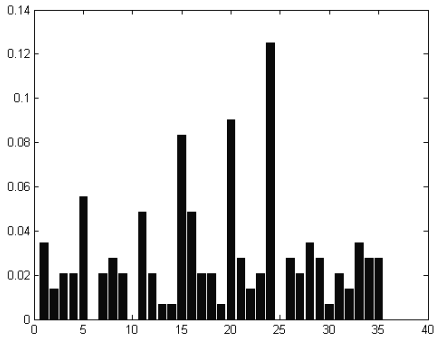
(c) cereal box



(d) flash light



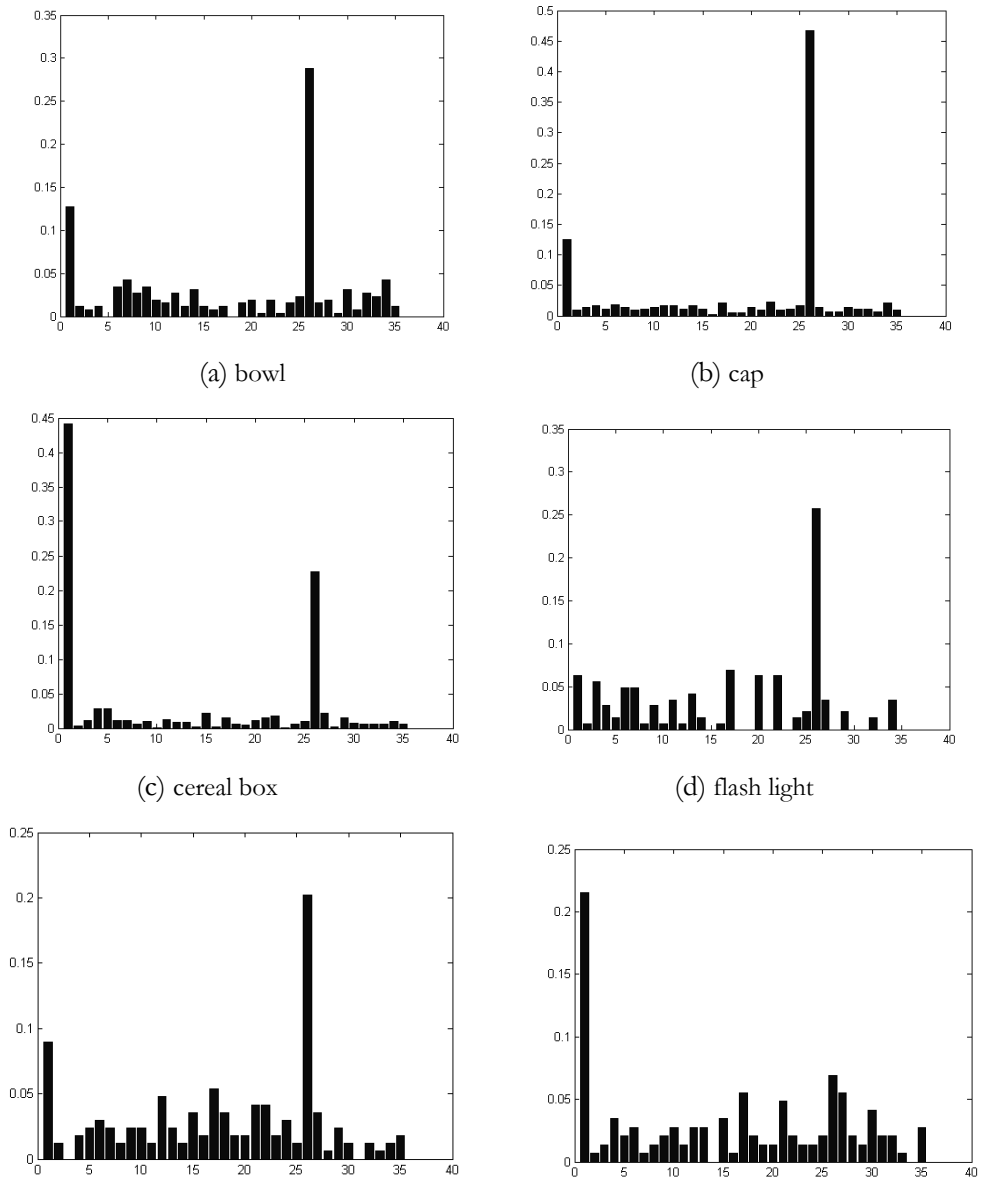
(e) Coffee mug_instance1



(f) coffee mug instance 2

Figure 4-17: Histogram representation of RGB gradient features over different objects, each histogram only represents one training image of an object

The similar histogram representation over the depth gradients feature could also be obtained using the same manner. The following chart is the respective result show in Figure 4-18, each histogram only represents one training image of an object



(a) bowl (b) cap

(c) cereal box (d) flash light

(e) coffee mug_instance1 (f) coffee mug instance 2

Figure 4-18: Histogram representation of depth gradients features over different objects

It is clear that the histogram distributions of each instance of objects are distinct, which made a fundamental basis for the further classification.

We further aggregated all the local features of every image to a global level. One of the above histogram representation was the global feature for one image , the value of each is later stored in one vector. For the convience and effeneicy of the later work. Finally,we aggregated the above vectors into one big vector with a size of 2332*35 which represent the entire feature vectors for all the training image. Here ,the 2332 is the number of the traininng image , and 35 is the dimension(variables) of the vector derived from the number of the visual words. In addition, no matter which type of feature descriptor, the generated histogram feature descriptor all have the same dimension and format. It make a basis for the later combination of different feature descriptor.

### 4.4.3. Parameter setting

In the bag of words workflow, three important steps were involved, during which the value of 2 parameters could affect the further recognition result.

#### 4.4.3.1. Sample number

First of all, in the sample section, we sampled n number of patch level feature vectors from feature vectors of every image. In other words, n random number of patch level vectors out of all the patch level vectors of one image were picked out. In the later computation of the centre vectors of the visual cluster, the value of the sampled vectors would affect the generation of the centre vectors (visual words). Therefore, it is required that the sampled feature vectors should be representative enough for the whole image features.

Here we had used the method of self test recognition assessment of training dataset mentioned in chapter 3 to optimize the value of the sample method. Take the size feature descriptor as the local feature, using qdc as the training classifier, and applied to our training dataset. Then in order to find the optimal value of the sample number, we change the value of the sample number from 10, 20 to 30. The respective accuracy results were acquired. (Show in the Table 4-4 and Figure 4-19). We finally find the optimized sample number to be 20.

Table 4-4: sample number with respective accuracy

| sample number | 10 | 20 | 30 |
|---|---|---|---|
| Category accuracy | 72.91% | 75.38% | 67.60% |
| Instance accuracy | 47.44% | 55.09% | 47.56% |

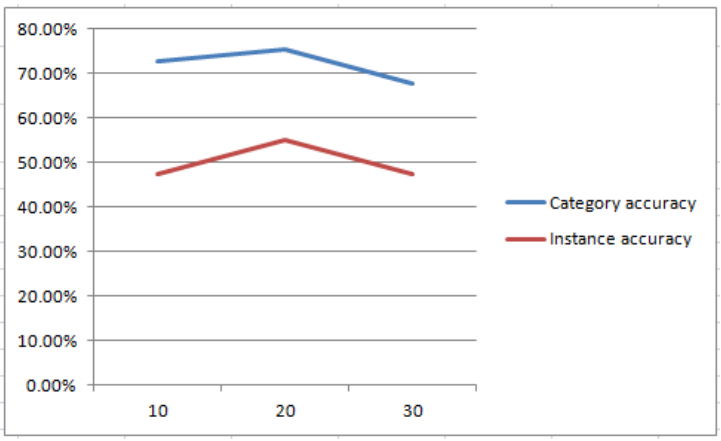



Figure 4-19: Sample number with respective accuracy

#### 4.4.3.2. Number of visual words

In addition, the number of the visual words is also matter the recognition result. To determine the number of visual words, we did a series of experiment. Take the size descriptor as the experiment. Following the same principle, we fixed the sampled number over the previous one and computed the classification accuracy using the self quality assessment manner. The final optimal number of visual words is 35. The respective results are displayed below:

Table 4-5: number of visual words with respective accuracy

| visual word number | 25 | 35 | 50 | 60 | 80 |
|---|---|---|---|---|---|
| Category accuracy | 69.43% | 75.38% | 70.06% | 70.32% | 72.38% |
| Instance accuracy | 43.70% | 55.09% | 55.65% | 51.85% | 43.77% |

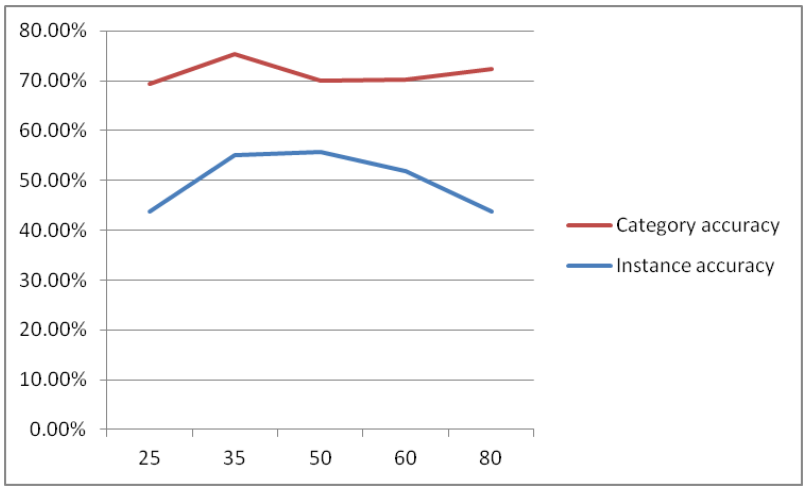



Figure 4-20: Number of visual words with respective accuracy

## 4.5. Classifier training

In this section, classifier training is a process that given sufficient training samples, specific classifier uses them to estimate the parameters of the discriminant function, as a result the discriminant function is determined when this process is over. Then this trained classifier can be used to classify the test images. As a pre-process, the feature vectors made in the previous step was associated with the corresponding true instance or category labels, and a training dataset is yielded. We first applied the qdc to our big training dataset, 8 categories of category training dataset and 19 instances of instance training dataset were made initially. The trained qdc classifiers on instance and category were then generated respectively. Later, due to the huge processing time of the svm, we sampled the big dataset into smaller one containing 5 categories of objects and totally 710 images, following the same principle, the trained svm classifier was produce later. A graph could illustrate here in the Figure 4-21:

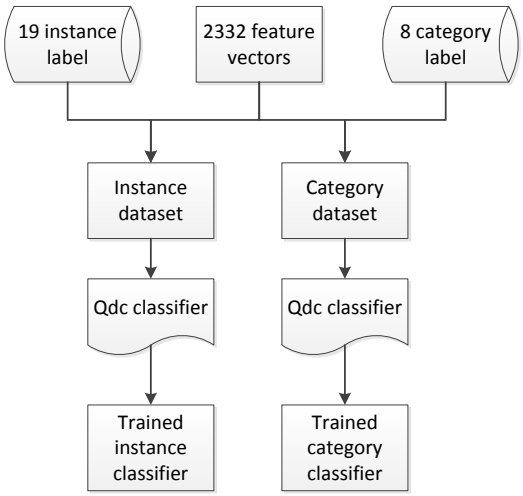



Figure 4-21: Training flow chart

## 4.6. Recognition over images of real scene

Our objective in the testing phase is that given an RGB image or depth image which contains an entire scene of daily objects, to recognize specific category or instance of the objects and locate the position of that object by certain visualisation manner. The testing RGB image and its corresponding depth image are illustrated in the Figure 4-22.

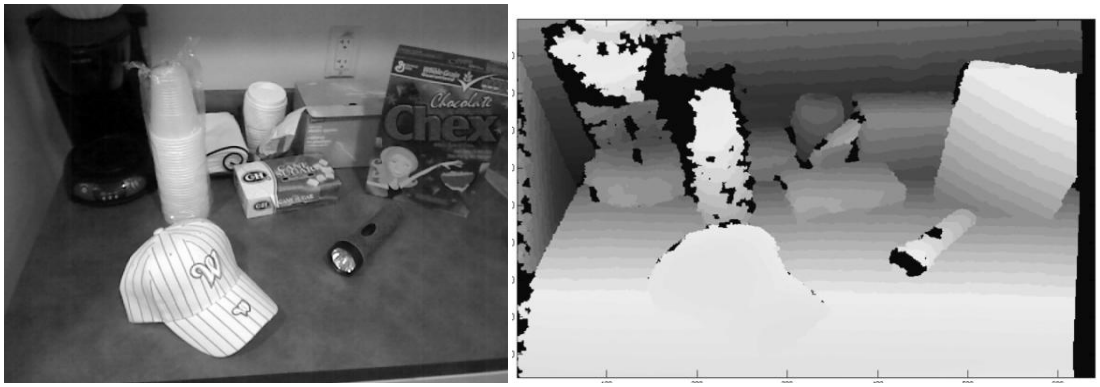



Figure 4-22: Test RGB image and corresponding depth image

First we aimed at distinguish caps from background and other objects. We cropped the testing image either RGB or depth into small overlapping windows with size of 210pixel* 140pixel. This size of window is defined similar with the size of the image of cap thus we could detect the cap effectively.

In the later section of multiple object recognition, the multiple size of the window is defined according to the size of the target objects we had in the training phrase. The image of real scene is finally partitioned into 918 windows. The cropped windows are illustrated in the Figure 4-23.

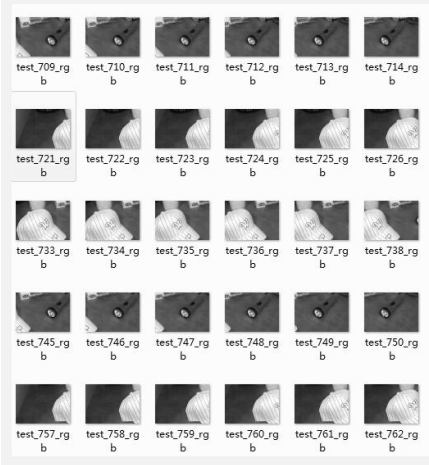



Figure 4-23: Partitioned windows from the test image

Further on, as the same way of feature extraction elaborated in the training section. The features of these windows were extracted using the same local feature descriptors and the pre-defined visual words. One entire feature vector of the windows was produced later with a size of a 918* 35.

A recognition dataset was made with 918 test samples, each sample has a feature vector of 35 dimensions. We could apply this testing feature dataset to the trained classifier. After then a dataset could be obtained. This dataset contained the posterior probability or the confidence level which indicated the presence of each objects.

Here we took the result sing the RGB gradient feature descriptor as the example. We selected several rows of this dataset as an example, see in the Table 4-6

Table 4-6 Dataset of posterior probability

| bowl | cap | Cereal box | Coffeemug1 | Coffeemug2 | Flash light |
|---|---|---|---|---|---|
| 0.112938 | 0.34898 | 0.007789 | 0.231992 | 0.251964 | 0.046338 |
| 0.100706 | 0.2671 | 0.025084 | 0.237268 | 0.24598 | 0.123861 |
| 0.081174 | 0.238848 | 0.151332 | 0.231622 | 0.225365 | 0.071659 |
| 0.067472 | 0.217205 | 0.13518 | 0.228817 | 0.226662 | 0.124663 |
| 0.06292 | 0.235222 | 0.04961 | 0.222833 | 0.257538 | 0.171877 |
| 0.043326 | 0.295266 | 0.086983 | 0.208362 | 0.326566 | 0.039498 |
| 4.35E-07 | 0.516858 | 0.423152 | 0.042197 | 0.01656 | 0.001232 |
| 3.17E-07 | 0.582137 | 0.302494 | 0.089655 | 0.023694 | 0.00202 |
| 6.37E-07 | 0.565757 | 0.203711 | 0.185232 | 0.041162 | 0.004137 |
| 2.10E-06 | 0.260456 | 0.117829 | 0.522114 | 0.086223 | 0.013376 |
| 0.000261 | 0.016056 | 0.016352 | 0.752235 | 0.146175 | 0.06892 |
| 0.000154 | 0.017335 | 0.015172 | 0.756546 | 0.153778 | 0.057015 |
| 6.79E-05 | 0.021382 | 0.009059 | 0.771473 | 0.156057 | 0.041962 |
| 2.69E-05 | 0.01793 | 0.009657 | 0.720088 | 0.212524 | 0.039774 |
| 1.12E-05 | 0.023225 | 0.006249 | 0.706326 | 0.232008 | 0.032181 |

The interpretation for the table is explained as below: Each specific column represents one instance of object, for example, the first column is bowl, second one denotes cap, etc. Each row represent one window .The values of those are the posterior probability or soft labels for each instance or category. Usually the larger the value, the more probability of the specific instance of object exists in that window. In addition, the sum of the value over one row is equal to one.

Also, we could see that in some rows, the value of cap is larger than others, but some are smaller. Those training objects did not appear in the above test image of scene could also a have a lager value in soft label. For example, in specific windows, the coffee mug which did not appear in the test scene has a value of 0.7 larger than others. This is due to the similar shape, size and gradient change with the plastic cup in the background.

Selection mechanism over the soft labels:
The target of the phrase is to find and locate the cap in the original image of real scene. Thereby, finding the windows that correctly represent the existence of the cap among the 918 windows is our objective here. We developed a selection mechanism to correctly find the right windows.
Here is the Pseudo code:

Loop every row of the dataset.
If value of (class 2)> value of (class1, 3, 4, 5, and 6):
Ratio= (Largest value in this row/second largest value in this row)>threshold value $\partial$

The Windows that satisfy the above condition are picked out and classified as cap.
After experiment, we optimized the threshold value $\partial$ best find and locate the windows: $\partial$=1.1

After applied this condition to the dataset, we acquired a number of bounding boxes (windows) which satisfied the condition above and classified as cap. An example of the windows being classified as cap can be seen in the Figure 4-24.

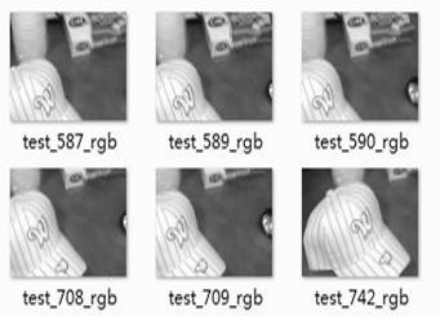



Figure 4-24: Windows classified as the cap

It is easy to locate the coordinates of the boundary and vertex of the windows in the original image, thus, an initial visualisation effect can be seen in the Figure 4-25(b). The mean value of the respective coordinates of the vertex of the box was calculated then and one bounding box is acquired. This final bounding box is our recognition result. (See in Figure4-25)

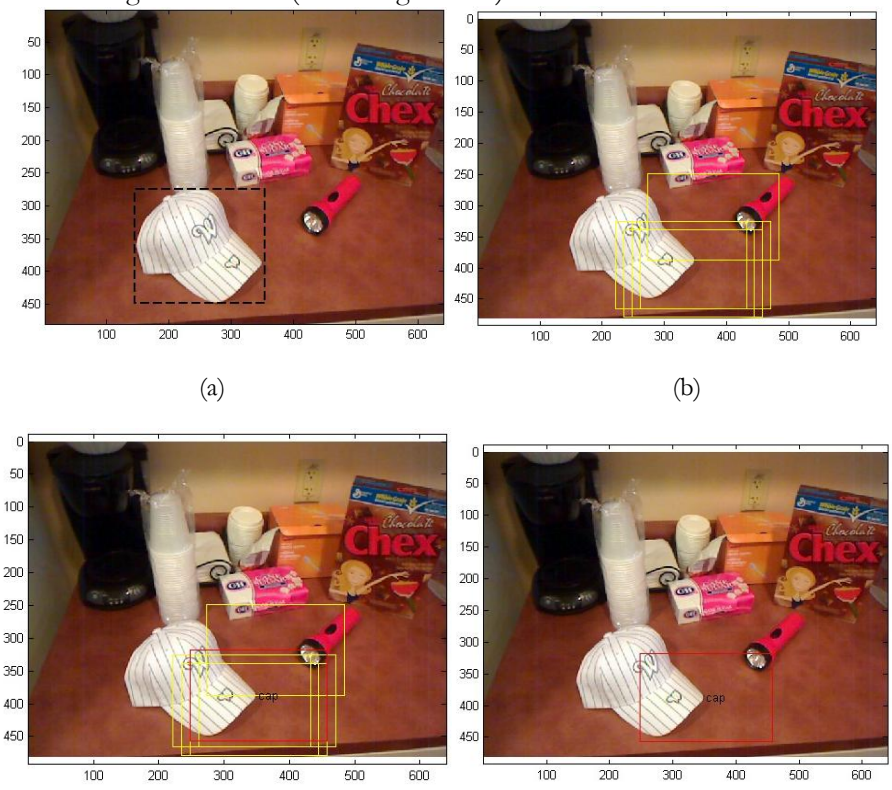
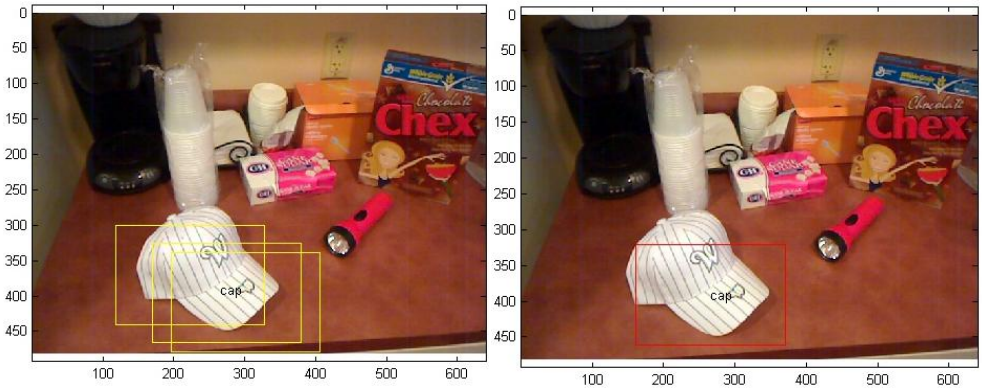



Figure 4-25: Visualisation image of the recognition result,(a)ground truth bounding box (b)initial windows classified as cap (c)mean bounding box over the windows previously. (d) Final recognition result.

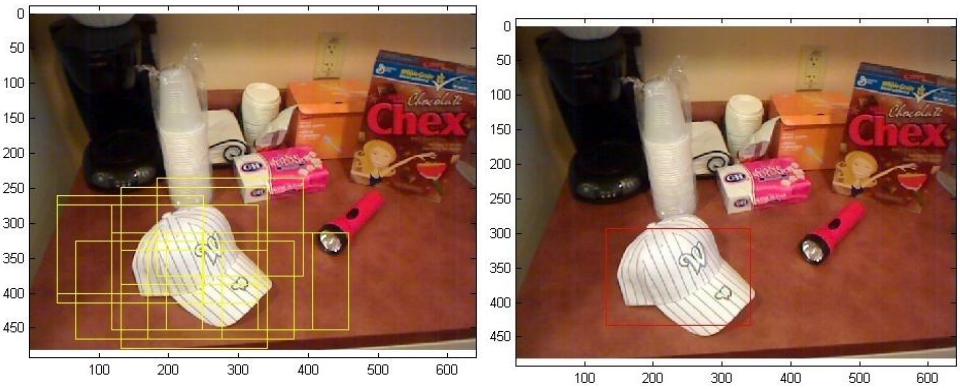
We also made the recognition over the same image using the size descriptors alone. And got respective result depicted in the Figure 4-26.




(a) (b)

Figure 4-26: Recognition result using size descriptor (a)original bounding box recognized as cap (b)final result

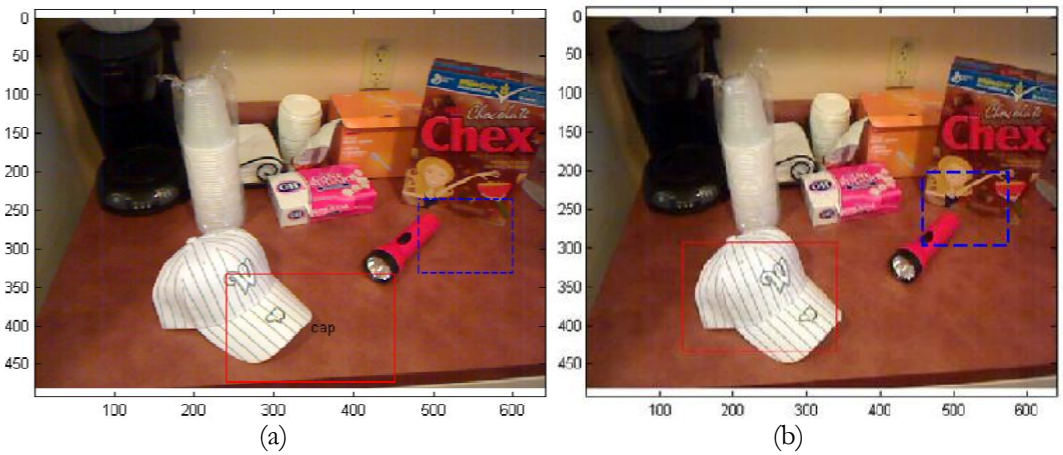
To see the effect of the combination of several features. We had experiment using the combination of the depth feature and RGB feature, which is illustrated below:



(a) initial windows classified as cap (b) Final recognition result.

Figure 4-27: Visualisation image of the recognition result over the combination of RGB image and depth image

Since the image of scene also contained the flash light which is also one type of object in our training dataset. We also did experiment which attempt to detect and recognize the flash light. Another size of sliding window was defined with as smaller size: 180*120. Using the same principle the respective features could be obtained. We use the combination of the RGB and detph features ,Together with the recognition of cap ,we could accomplish the following result,depetied in the Figure 4-28.




(a) (b)

Figure 4-28: Multiple object recognition result the red bounding box refer to the cap, the blue one denote the flashlight (a) recognition using RGB descriptor (b) recognition using the combination of RGB and depth features

It could see form the above graph that results over multiple object recognition is not very ideal.

we had also made recognition experiment to recognize other object over other images of real scene. The recognition result of coffemug_1 , see Figure 4-29:

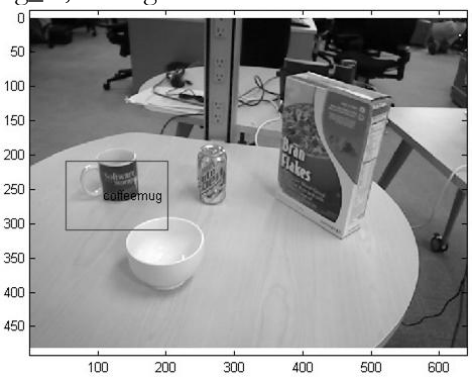



Figure 4-29: Coffee mug recognition result

# 5. QUALITY ASSESSMENT AND DISCUSSION

## 5.1. Self test recognition assessment and analysis

For the aim of evaluating the quality over the previously seen and trained data, a self test quality assessment of object recognition was made.

### 5.1.1. Evaluation over big dataset using trained qdc

Here we use a bootstrapping method to create the dataset for training and dataset for evaluation based on the original training dataset. The principle of the bootstrapping is described below: First randomly take a sample, put it in the test dataset and then return it to the training dataset. Repeat this operation m<n times. Then we have a training set of n samples and an evaluation set of m samples. The result after the bootstrapping step is that we got 2332 samples as the training set and 768 for the evaluation set. The qdc classifier was trained using 2332 training samples and tested on the 768 evaluation set.

To be noted, either training or evaluation set all have the same form: one image only contain one object and the background is removed.

A prediction of the recognized instance or category label of the evaluation set could be acquired by applying the trained classifier to the evaluation set, the error estimation was given by comparing the prediction label of the evaluation dataset with the true label of the corresponding samples. In other words, this self test aims at evaluating the result of recognition over the known data which only contain one object per image.

Several evaluation metrics were considered in this section:
TP (True positive): the number of images correctly classified as the corresponding objects.
N: total number of the evaluation set samples.
OA (Overall accuracy): the ratio of the number of correctly classified samples to the total number of the evaluation samples.

$$OA=TP/N \quad (5\text{-}1)$$

We applied this overall accuracy metric over instance recognition and category recognition. The feature descriptor we developed can be tested separately or jointly. The evaluation result is illustrated in the Table 5-1 and Figure 5-1

Table 5-1: overall accuracy on different descriptors over big dataset using qdc

| descriptor | Instance accuracy | Category accuracy |
|---|---|---|
| RGB gradient | 78.97% | 94.28% |
| Depth gradient | 54.25% | 85.70% |
| size descriptor | 55.09% | 75.38% |
| RGB gradient+ depth gradient | 56.48% | 93.3% |
| 3feature together | 50.87% | 87.72% |

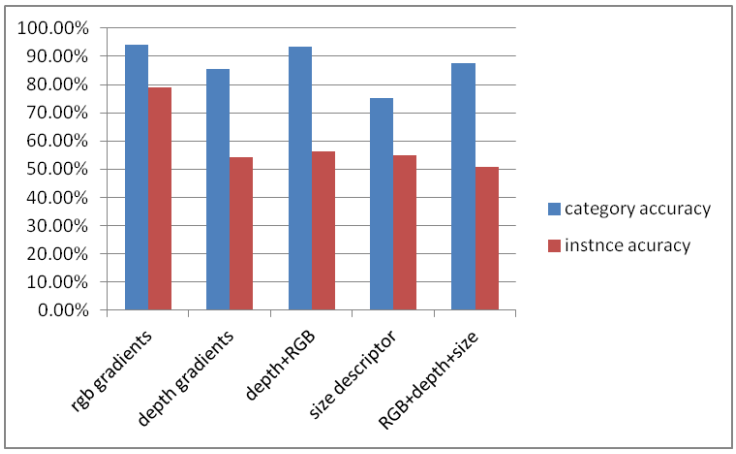



Figure 5-1: Overall accuracy on different descriptors

As illustrated in the above table and figure, we could see that, the category recognition accuracy over either of the descriptors or the combination of them is much better than that of instance recognition accuracy. Specifically, the performance of the RGB gradients over two types of recognition is better than other descriptors or the combination. The performance for the combination of these three descriptors has indicated a better result than the depth alone and size alone but lower than that of the RGB descriptor. It seems that result of the combination of either two descriptor or three descriptors is the average of the sum of the each descriptor.

### 5.1.2. Evaluation over small dataset using trained svm

Since we have a reduced dataset which only contained 751 images of 5 categories and 6 instances of objects, we also tested the overall accuracy using svm as the classifier.
Following the same principle, only except that we use the small dataset sampled from big dataset. Here are the respective accuracy results:

Table 5-2: overall accuracy on different descriptors over small dataset using svm

| | RGB gradients | depth gradients | size descriptor | depth+RGB | RGB+depth+size |
|---|---|---|---|---|---|
| category accuracy | 91.24% | 80.84% | 84.86% | 92.86% | 98.13% |
| instance accuracy | 86.99% | 76.36% | 71.17% | 90.49% | 95.83% |

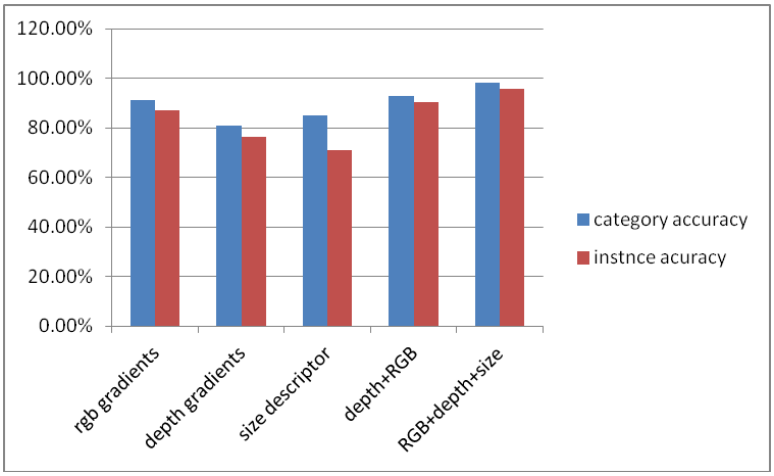



Figure 5-2: Overall accuracy over the small dataset

In this experiment, we could see that for both of the individual descriptors or the combination, the result of category recognition only has a slight superiority than that of the instance recognition.
In addition, the combinations of two descriptors or three descriptors all perform a better recognition result than any individual features only. It can be explained by the character of the svm. The svm could balance the weight over each individual descriptor by supervised learning, thus the significance of each descriptor could appear when they combined together.

In terms of the result of the individual descriptor, the RGB gradient is still on the top, followed by depth and size descriptor. Another observation is that compared with the result over big dataset. The performance over instance recognition is getting much better with a significantly increase. While on the other hand, the performance over category recognition basically keeps the same or even lower in some case. It could be the reason that in the previous big dataset, every category contains more number of sample images than that of the instance. But in the small dataset this difference is not obvious.

We could also see from this graph, that the size descriptor over category recognition has an obvious large superiority performance than that of instance recognition. Because each category of objects has particular size character, while for several instance of one category could similar in size.

We also computed the confusion matrix of the evaluation over several descriptors (see in Table 5-2). The labels in the tables are corresponding to 1 bowl ,2 cap ,3 cereal box, 4 coffee mug_1 , 5 coffee mug_2 ,6 flashlight. The labels of the category recognition are: 1 bowl ,2 cap ,3 cereal box, 4 coffee mug,5 flashlight.

Table 5-3 : confusion matrix of the overall accuracy

(1) Size feature descriptor over instance recognition

| True Labels | Estimated Labels 1 | 2 | 3 | 4 | 5 | 6 | Totals |
|---|---|---|---|---|---|---|---|
| 1 | 41 | 3 | 0 | 0 | 0 | 0 | 44 |
| 2 | 23 | 27 | 0 | 0 | 0 | 0 | 50 |
| 3 | 0 | 0 | 49 | 0 | 0 | 0 | 49 |
| 4 | 0 | 1 | 0 | 2 | 30 | 11 | 44 |
| 5 | 0 | 0 | 0 | 0 | 37 | 6 | 43 |
| 6 | 0 | 3 | 0 | 4 | 0 | 44 | 51 |
| Totals | 64 | 34 | 49 | 6 | 67 | 61 | 281 |

(2)Three features together_ instance recognition

| True Labels | Estimated Labels 1 | 2 | 3 | 4 | 5 | 6 | Totals |
|---|---|---|---|---|---|---|---|
| 1 | 41 | 0 | 0 | 0 | 0 | 0 | 41 |
| 2 | 0 | 54 | 0 | 0 | 0 | 0 | 54 |
| 3 | 0 | 0 | 46 | 0 | 0 | 0 | 46 |
| 4 | 0 | 0 | 0 | 45 | 3 | 1 | 49 |
| 5 | 0 | 0 | 0 | 5 | 37 | 0 | 42 |
| 6 | 0 | 1 | 0 | 2 | 0 | 40 | 43 |
| Totals | 41 | 55 | 46 | 52 | 40 | 41 | 275 |

(3)3 features together category recognition

| True Labels | Estimated Labels 1 | 2 | 3 | 4 | 5 | Totals |
|---|---|---|---|---|---|---|
| 1 | 45 | 0 | 0 | 0 | 0 | 45 |
| 2 | 0 | 45 | 0 | 0 | 0 | 45 |
| 3 | 0 | 0 | 41 | 0 | 0 | 41 |
| 4 | 0 | 0 | 0 | 81 | 0 | 81 |
| 5 | 0 | 0 | 0 | 7 | 42 | 49 |
| Totals | 45 | 45 | 41 | 88 | 42 | 261 |

From the table (1) of Size feature descriptor over instance recognition elevation, we can see that the size feature does not distinguish well between coffee mug_1 and coffee mug_2, which conform to our expectation that the size is weaker in distinguishing the instances from the same category. As the result over the combination of three feature descriptors, a few misclassifications are found.

From the above experiment, we could draw the following conclusion: First, in the self test recognition, if only consider the performance of individual descriptor over both classifiers, RGB gradient features over the RGB data from Kinect have the best performance, followed by depth gradients and size features. Second, Using svm as the classifier, the combination of three features have the best performance than any other individual descriptors. Whereas in the case of qdc, it seems that the accuracy result over the combination of several features average the sum of that of individual features.

## 5.2. Assessment over recognition image of scene

### 5.2.1. Assessment result

In this section, since we had obtained one final window which classified as target object initially. Compared with the ground truth bounding box defined by ourselves with visual inspection, the classified result could be assessed in the similar manner described above. Two major evaluation methods could be applied to the recognition based problem: one is pixel based approach, the other is object based approach. However, in our case the object based manner is not suitable since the number of object in one image is limited. And we only did experiment to recognize one or two objects in the Kinect data.

Since our recognition was finally visualized in RGB colour image, It is consists of pixels. According to the (Smirnov et al., 2006), a pixel based recognition manner was adopted and several metric were considered to assess the performance of our recognition work on data of real world scene:
TP (True positive): number of pixels that correctly classified as the corresponding objects
FP (false positive): number of pixels that background or other objects classified as target object.
Overall accuracy: $OA=TP/(TP+FP)$

We had defined the ground truth bounding box that tightly surrounds the target object, illustrated in the Figure 5-3(a).the pixels that should count as the TP is the intersection region between ground truth box and the window classified as target object, see Figure5-3(b). The pixels that should count as FP is the remaining area except of the intersection part of the red bounding box.


(a) (b)

Figure 5-3: Illustration of ground truth box and window being recognized (a) the black bounding box denote the ground truth box(b)red box is the window classified as cap

In the experiment of cap recognition, our goal is to recognize cap from the test image. Following the above assessment metric, we had computed the above metric over the recognition work using RGB

gradient and the overall accuracy of the cap recognition is 48.04%, following the same manner, we could get the result of other descriptor. A result table is showed below:

Table 5-4: cap recognition overall accuracy over different descriptor

| descriptor | RGB gradients | Depth gradients | Size descriptor | RGB+Depth+size |
|---|---|---|---|---|
| Recognition accuracy | 48.04% | 77% | 70% | 85% |

As illustrated in the above table, the performance of depth descriptor is better than the size one, followed by size descriptor. And the combination of the three features has the top accuracy.

The reason could be that the depth features capture the real outer edge and boundary of the object, which has a strong gradients detection of the object in the clutter scene. While the RGB one can also capture the strong colour gradient change of the object, but objects which has similar texture and colour character with the cap would confuse the recognition. For example in the recognition image, the white plastic cup has the similar texture change with the cap, which would have a negative impact over the recognition of the cap. In addition, since we had used the svm as the training classifier. Proved in self test section, the svm could well balance the advantage of each descriptor thus their performance is better than the individual one.
Due to the limitation of time, we did not make the accuracy assessment over the multiple object recognition result.
More number of experiments is needed in further work to help us analysis the weak and strength of our proposed method.

### 5.2.2. Result analysis and discussion

We had made cap and coffee mug recognition test over image of real world data using different descriptors. The experiment had indicated that using RGB gradients alone, depth and size alone could basically locate and recognize the cap. To be noted, the combination of them performed a better result.
However when it comes to multiple object recognition over the same image, the recognition did not perform a very satisfactory result. Misrecognitions were found in our experiment.
Several reasons could lead to the low accuracy of the multiple object recognition.

#### 5.2.2.1. Error sources from data

Start with the problem of the RGB-D data, Firstly, the image of real scene contains cluttered background, table and other objects which confuse the recognition. This information on background, table and objects which did not appear in our training set would contribute to a negative impact over the posterior probability of the target object in the partitioned windows. Besides, according to (Khoshelham et al., 2012) Due to the character of the Kinect sensor, the depth image of the scene contains a lot of gaps and blank area. Systemic noisy could also be found. In addition, the sensor would fail or capture a little depth measurement over specific material such as transparent glass or some surface of the metal. Thus, our depth gradient feature over depth image could not capture information over this area of the object appeared in the scene image. In addition, the point clouds which converted from the depth image would have more gaps, as illustrated in the Figure 5-1. Moreover, some of the target object is behind other objects and occluded by them in the image of real scene. The occlusion could influence the recognition since the features over the object is not complete. In our work, the size features over point cloud would be influenced most at this point. Because the point clouds of the object is incomplete which could not give sufficient and comprehensive distance measurement of the objects. All in all, this inconsistency with the features over the training dataset would make it difficult for the later similarity measure and recognition.

Figure 5-4: Point clouds of the image of real scene

#### 5.2.2.2. Error sources from method

In terms of our recognition method, one drawback of our method which takes some responsible for the inaccurate of the recognition is that this partitioned window based classification. The size of the sliding window is changing each time depends on the size of the object in the training image. Nevertheless, with the change of the scale and shift of recognition image, the object to be recognized varies over size. Therefore, the window to be detected and recognized contains a lot of irrelevant information, which could interfere the recognition.

Another point is that our adopted size feature descriptor is not very robust and accurate enough to capture the size feature over the point clouds. The distance measurement is too simple to capture and calculate the parameters that matters the size of the objects.
The selection mechanism we developed could not apply well to a general application, since the threshold value is optimized by experiment. Therefore, incorrect detection of the window could be made. As a consequence, the corresponding recognition would be imprecise.

#### 5.2.2.3. Error sources from assessment

Our pixel based assessment over recognition has some defect over our recognition experiment. First of all, As the reference ground truth, the manually defined bounding box might not have the same size with that of sliding window, which could lead error to computation of overall accuracy. Besides, the number of our experiment is not sufficient enough to give a comprehensive statistics estimation.

# 6. CONCLUSION AND RECOMMENDATIONS

## 6.1. Conclusion

In this research, we had developed a methodology on indoor object recognition using RGB-D data acquired by Kinect. The Kinect and similar low price RGB-D camera had demonstrated a significant potential over the indoor perception problems such as object recognition, indoor mapping, etc. however Due to the limitation of the indoor environment, indoor perception work has to withstand limited lighting, insufficient features, and changing structures. Visual based approach using traditional optical camera had shortcomings of lack sufficient geometric information and need huge time in matching pictures. While on the other hand, laser scanning approach is expensive and requires extra time to register the visual data. In our research, the combination of utilizing features over RGB data and depth data has proved a better performance than the individual approach alone.
Some conclusion over the research is summarized below:

1) The adoption of the bag of word concept significantly enhanced the recognition performance. It aggregated those local features into a global image level with the same histogram representation over all the images and unified the format of different local descriptors, which make a basis for further combination of three features. In addition, the sample manner and K-mean clustering which was the sub-procedure of that had significantly enhanced the efficiency and accuracy of the generation of the visual words. We have optimized the number of visual words to be 35 and sample number per image to be 20 in the bag of words model, which had achieved the best performance over our training data.

2) In the experiment of self test recognition over data which contain one object per image without background. The performance of RGB gradient was leading than other two, followed by depth gradient features and size descriptor. For all the descriptors, the accuracy over the category recognition is better than that of the instance recognition due to the larger number of training samples. Thus a larger number of training samples per class could improve the recognition performance.

3) In the case of recognition over image which contain clutter and multiple objects, the result of one object recognition was accepted, while in terms of multiple object recognition, our method was not robust and could not achieve a satisfied result.

4) In both task of recognition, the combination of these three local features over RGB, depth and point clouds had beyond the performance than any individual features if using svm as the training classifier. The use of the depth image and respective point clouds has explored more shape and size character of the objects, which could make a complements for the use of traditional RGB image over the recognition.

5) The qdc is superior in training time but did not show a robust ability in combining the advantage over several features together. While the svm classifier cost a longer time for training and could yield relative accurate posterior probability over each unlabeled class. What is more, it performed a strong

ability of balancing the importance of different descriptors and the result of the combination of descriptors shown a better result than the individual descriptor alone.

### 6.2. Answers to the research questions:

1) Which local features or representations of the object are important and effective for our recognition work?

In this work, we had developed three local feature descriptors to extract three types of features: gradient features over RGB image, gradient features over depth image, size features over point clouds image. The result had indicated that using qdc as the training classifier, the RGB gradient performed to be more effective for recognizing objects, whereas using svm, the combination of the three features were more effective.

2) How to detect and extract the features from the measured data in an efficient way?

A patch based manner was adopted to transform the pixel level features into patch level features initially. Then, bag of words quantization and frequency description was introduced to aggregate the local features into global level.

3) How to aggregate these local features extracted in the previous step in a global level?

The bag of words concept provides a method to arrange the feature vectors in clusters and later aggregated the feature vectors of all the training images in one vector.

4) How to combine the colour (RGB) features and depth (D) features in the feature extraction?

Before the training section, three separate features over RGB and depth were extracted and stored. We could combine them by stacking the feature vectors into one vector. And later the respective training dataset could be generated.

5) How to match the features extracted in the previous step with the corresponding features in the existing training dataset in an efficient way?

In our work, we utilized the classifier based approach. Two classifiers were adopted: svm and qdc. These classifiers were trained by our training samples which is the representation of the features over the training dataset, so that the parameters of the discriminant function over each classifier could be estimated and determined when this training process is over. As a result, these trained classifiers with the prior knowledge can be used to classify the test image.

6) How to keep recognition work invariant to rotation, shift or change of scale?

In our adopted approach, the image for one object either RGB or depth in the training dataset was all displayed in a multi-view way which were sampled from a continuous frame of the image recording. Later the features over these multi-view images were trained using classifier. This manner had guaranteed the recognition invariant to rotation. Moreover, since the position of object is not fixed in different image the sliding window approach we adopted could detect and locate the position of the target object in the image of scene regardless of the change of the object, which keep an invariant recognition to the shift of object in the image of scene. Our method is currently not robust in recognition over the change of scale.

7) How to assess the quality of our result?

Two quality assessments were involved. First, to assess the quality of the trained classifier, an error estimation of the prediction result over the trained classifier was carried out.

Secondly, to assess the accuracy of the final recognition result, this pixel based classification metric was considered.

8) What is the advantage of RGB-D data in comparison with colour-less point clouds or images?

Compared with the color-less point clouds, RGB-D data has one more colour channel. Since the point clouds can hardly capture the texture features over the surface of the object, and due to the weakness and low efficiency in constructing 3d geometric data, the colour image is not robust in providing shape or size information over the object. The RGB image could provide texture information over the recognition, while the combination of them complements each other. In our recognition work, it has been indicated that using proper classifier, the combination of features over two channels could perform a better result than that of the individual one.

### 6.3. Recommendations

1) Our final recognition result over the real world image was not very ideal and robust. The partitioned windows contained cluttered background and some other irrelevant objects which lead to confusion over the classifier and recognition. Thus further work could be done to improve the method in the final recognition over the images of the real world scene. For example utilizing the depth image, we could make a point clouds over the test image and introduce a well adapted segmentation technique. Further recognition based on the segments of the test image could be more accurate and robust.

2) In our experiment, we only made one or two object recognition over the test images. More objects and simultaneous multiple objects recognition of experiment is needed and respective revise of the method and selection mechanism should be developed.

3) Our exploration of the use of the depth image over recognition work only includes two feature descriptors. More robust and effective feature descriptor could be studied and applied. Moreover, our proposed size feature has shown a potential of using point clouds features in the recognition. More feature descriptor could be applied to extract the features over the point clouds. And more data processing work on point clouds can be further explored to enhance the recognition, like segmentation, or point clouds reconstruction using depth image matching.

4) The proposed method only attempted to work on small daily objects, more work should be done to recognize the big indoor objects and indoor furniture using RGB-D data. In addition, after the success of the recognition section, further work could be done on object modeling and indoor modeling

5) There are shortcomings for the developed bag of words approach. On one hand, the feature vectors over one image varies one to another, thus the number of visual words (cluster) affects the further performance and accuracy so that experiment is needed to optimize this value. On the other hand, since the final histogram representation separated the feature vectors into numbers of bins, the value of bin boundary could yield problems on matching between flat histograms. A pyramid matching approach(Lazebnik et al., 2006) could be adopted to overcome the weakness and enhance the recognition performance.

6) The whole process of feature extraction, classifier training, and final testing on data of real world scene is not automatic without any human intervention. Times of experiments are needed to set the optimized parameters and threshold value in the program. Further work should be done to improve the automation of the system and a better self learning ability. Besides, since our program is written in Matlab, and the running time of the program is not very fast. The optimization of the program or

rewrite in C++ could be done, so that our recognition of the real scene can be more efficiency and faster.

# LIST OF REFERENCES


Beis, J. S., & Lowe, D. G. (1997). *Shape Indexing Using Approximate Nearest-Neighbour Search in High-Dimensional Spaces*. Paper presented at the Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97).

Bo, L., Lai, K., Ren, X., & Fox, D. (2011). *Object recognition with hierarchical kernel descriptors*. Paper presented at the Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition.

Bo, L., Ren, X., & Fox, D. (2010). Kernel descriptors for visual recognition. *Advances in Neural Information Processing Systems*.

Campbell, R. J., & Flynn, P. J. (2001). A survey of free-form object representation and recognition techniques. *Comput. Vis. Image Underst., 81*(2), 166-210. doi: 10.1006/cviu.2000.0889

Canny, J. (1986). A Computational Approach to Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8*(6), 679-698. doi: 10.1109/tpami.1986.4767851

Collet, A., Srinivasa, S. S., & Hebert, M. (2011, 9-13 May 2011). *Structure discovery in multi-modal data: A region-based approach*. Paper presented at the Robotics and Automation (ICRA), 2011 IEEE International Conference on.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273-297. doi: 10.1007/bf00994018

Du, H., Henry, P., Ren, X., Cheng, M., Goldman, D., Seitz, S., & Fox, D. (2011). *Interactive 3D Modeling of Indoor Environments with a Consumer Depth Camera*. Paper presented at the 13th International Conference on Ubiquitous Computing (Ubicomp 2011, Beijing, China.

Fergus, R. (2008). Visual Object Recognition and Retrieval, from http://cs.nyu.edu/~fergus/icml_tutorial/

Frome, A., Huber, D., Kolluri, R., Bülow, T., & Malik, J. (2004). Recognizing Objects in Range Data Using Regional Point Descriptors. In T. Pajdla & J. Matas (Eds.), *Computer Vision - ECCV 2004* (Vol. 3023, pp. 224-237): Springer Berlin Heidelberg.

Gevers, T., & Smeulders, A. (1997). Color based object recognition

Image Analysis and Processing. In A. Del Bimbo (Ed.), (Vol. 1310, pp. 319-326): Springer Berlin / Heidelberg.

Grauman, K., Shakhnarovich, G., & Darrell, T. (2003, 13-16 Oct. 2003). *Inferring 3D structure with a statistical image-based shape model*. Paper presented at the Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on.

Hao, D., Peter, H., Xiaofeng, R., Marvin, C., Dan, B. G., Seitz, S. M., & Dieter, F. (2011). *Interactive 3D Modeling of Indoor Environments with a Consumer Depth Camera*. Paper presented at the 13th international conference on Ubiquitous computing. http://ai.cs.washington.edu/www/media/papers/paper_3.pdf

Hays, J. (2011). Local Features and Bag of Words Models, from http://cs.brown.edu/courses/cs143/lectures/16.pdf

Henry, P., Krainin, M., Herbst, E., Ren, X., & Fox, D. (2012). RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *The International Journal of Robotics Research, 31*(5), 647-663. doi: citeulike-article-id:10583788

doi: 10.1177/0278364911434148

Hetzel, G., Leibe, B., Levi, P., & Schiele, B. (2001, 2001). *3D object recognition from range images using local feature histograms*. Paper presented at the Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on.

Jinfu, Y., Kai, W., Ming-ai, L., & Lu, L. (2011, 7-10 Aug. 2011). *Research on object recognition using bag of word model for mobile robot navigation*. Paper presented at the Mechatronics and Automation (ICMA), 2011 International Conference on.

Johnson, A. E., & Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 21*(5), 433-449. doi: 10.1109/34.765655

Kanezaki, A., Harada, T., & Kuniyoshi, Y. (2010). Partial matching of real textured 3D objects using color cubic higher-order local auto-correlation features. *Vis. Comput., 26*(10), 1269-1281. doi: 10.1007/s00371-010-0521-3

Kang, M., & Kim, J. (2007). Real Time Object Recognition Using K-Nearest Neighbor in Parametric Eigenspace. In K. Li, M. Fei, G. Irwin & S. Ma (Eds.), *Bio-Inspired Computational Intelligence and Applications* (Vol. 4688, pp. 403-411): Springer Berlin Heidelberg.

Kevin, L., Bo, L., Ren, X., & Fox, D. (2011). *A large-scale hierarchical multi-view RGB-D object dataset.* Paper presented at the ICRA.

Khoshelham, K. (2007). Extending generalized hough transform to detect 3D objects in laser range data. *In: Proceedings of the ISPRS workshop Laser Scanning 2007 and SilviLasser 2007, Espoo, Finland, 12-14 September 2007 / ed. by P. Rönnholm, H .Hyyppä and J. Hyyppä. (ISPRS : Volume XXXVI, part 3/W52. pp.206-210.*

Khoshelham, K., & Elberink, S. O. (2012). Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors, 12*(2), 1437-1454.

Kobayashi, T., & Otsu, N. (2004). *Action and Simultaneous Multiple-Person Identification Using Cubic Higher-Order Local Auto-Correlation.* Paper presented at the Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4 - Volume 04.

Lai, K., Liefeng, B., Xiaofeng, R., & Fox, D. (2011, 9-13 May 2011). *A large-scale hierarchical multi-view RGB-D object dataset.* Paper presented at the Robotics and Automation (ICRA), 2011 IEEE International Conference on.

Lazebnik, S., Schmid, C., & Ponce, J. (2006, 2006). *Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories.* Paper presented at the Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on.

Lin, Z., Wang, J., & Ma, K.-K. (2002). Using eigencolor normalization for illumination-invariant color object recognition. *Pattern Recognition, 35*(11), 2629-2642. doi: http://dx.doi.org/10.1016/S0031-3203(01)00207-2

Lowe, D. G. (1999, 1999). *Object recognition from local scale-invariant features.* Paper presented at the Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on.

Mian, A. S., Bennamoun, M., & Owens, R. (2006). Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 28*(10), 1584-1601. doi: 10.1109/tpami.2006.213

Microsoft.). Kinect. Retrieved 14 December, 2011, from http://www.xbox.com/en-us/kinect/

Mikolajczyk, K., & Schmid, C. (2004). Scale \& Affine Invariant Interest Point Detectors. *Int. J. Comput. Vision, 60*(1), 63-86. doi: 10.1023/B:VISI.0000027790.02288.f2

openkinect.). from http://openkinect.org/wiki/Imaging_Information

Pavel, F. A., Zhiyong, W., & Feng, D. D. (2009, 5-7 Oct. 2009). *Reliable object recognition using SIFT features.* Paper presented at the Multimedia Signal Processing, 2009. MMSP '09. IEEE International Workshop on.

Pontil, M., & Verri, A. (1998). Support vector machines for 3D object recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20*(6), 637-646. doi: 10.1109/34.683777

PrimeSense.). from http://www.primesense.com/

Prince, S. J. D. (2012). *Computer Vision*: Cambridge University Press.

Rubner, Y., Tomasi, C., & Guibas, L. J. (2000). The Earth Mover's Distance as a Metric for Image Retrieval. *Int. J. Comput. Vision, 40*(2), 99-121. doi: 10.1023/a:1026543900054

Rusu, R. B., Blodow, N., & Beetz, M. (2009). *Fast point feature histograms (FPFH) for 3D registration*. Paper presented at the Proceedings of the 2009 IEEE international conference on Robotics and Automation, Kobe, Japan.

Scholkopf, B., Burges, C. J. C., & Smola, A. J. (1999). *Advances in Kernel Methods: Support Vector Learning*: Mit Press.

Silverman, B. W., & Jones, M. C. (1989). E. Fix and J.L. Hodges (1951): An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951). *International Statistical Review / Revue Internationale de Statistique, 57*(3), 233-238. doi: 10.2307/1403796

Smirnov, E. N., & Kaptein, A. (2006, Dec. 2006). *Theoretical and Experimental Study of a Meta-Typicalness Approach for Reliable Classification.* Paper presented at the Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on.

The Standard SVM Formulation.). from http://research.microsoft.com/en-us/um/people/manik/projects/trade-off/svm.html

Tangelder, J. W. H., & Veltkamp, R. C. (2004, 7-9 June 2004). *A survey of content based 3D shape retrieval methods.* Paper presented at the Shape Modeling Applications, 2004. Proceedings.

Text Classification in Python.). from http://www.python-course.eu/text_classification_python.php
Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence, 99*(1), 21-71. doi: 10.1016/s0004-3702(97)00078-7
Tombari, F., & Stefano, L. D. (2012). Hough Voting for 3D Object Recognition under Occlusion and Clutter. [Research paper]. *IPSJ Transactions on Computer Vision and Applications, Vol.4 1–10 (Mar. 2012)*. doi: 10.2197/ipsjtcva.4.1
Ulrich, M., Steger, C., & Baumgartner, A. (2003). Real-time object recognition using a modified generalized Hough transform. *Pattern Recognition, 36*(11), 2557-2570. doi: 10.1016/s0031-3203(03)00169-9
Vandevenne, L. (2004). Lode's Computer Graphics Tutorial, from http://lodev.org/cgtutor/filtering.html
Willems, G., Tuytelaars, T., & Gool, L. (2008). *An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector*. Paper presented at the Proceedings of the 10th European Conference on Computer Vision: Part II, Marseille, France.