CLOUD SERVICE FOR RECOGNITION OF SPECIFIC 3D OBJECTS

by

Apiporn Simapornchai

A thesis submission in partial fulfillment of the requirements for the degree of Master of Science in Information Management

Examination Committee: Dr. Matthew N. Dailey (Chairperson)
Dr. Raphael Duboz
Dr. Erik L.J. Bohez

Nationality: Thai
Previous Degree: Bachelor of Science in Information Technology
Stamford International University, Thailand

Scholarship Donor: AIT Fellowship

Asian Institute of Technology
School of Engineering and Technology
Thailand
July 2013
Acknowledgment

My thesis work was accomplished through the assistance and support of various people. I would like to take this opportunity to give my appreciation here.

Sincere appreciation to my advisor, Matthew N. Dailey, who offered me valuable advices and encouragement throughout the study. I would not have been able to make it without him. In addition, I would like to extend my sincere gratitude to my committee members, Raphael Duboz and Erik L.J. Bohez, for their suggestions and constructive criticism.

I would like to express my thanks to The Asian Institute of Technology, especially the School of Engineering and Technology, for its support and assistance during my studies at AIT. Moreover, Rangsit University has been a major supporter of the research tools necessary for this research, thus I am truly thankful for its kindness.

Last but not least, great gratitude to Mr. Parkpoom Chaisiriprasert for providing the significant amount of support that I needed to complete this research. A special word of appreciation and gratitude goes to my beloved parents, for their inspiration and for giving me endless unconditional love.

I dedicate this piece of work to all these people mentioned above.
Abstract

Due to constraints on cost, power consumption, perceptual capabilities, knowledge management, reasoning capabilities, and compute power, the adoption of advanced service robots has thus far been limited. In order to spread the adoption of service robots, further development is crucial. This thesis is study on the use of cloud computing platforms that support robotic applications, specifically object recognition, expanding the compute compute and storage capacity of the robot system. The thesis presents a system architecture, implemented prototype, and initial experimental data on a cloud infrastructure platform for recognition of specific 3D objects. The objective is to prove the concept of enabling robot to share knowledge with others via cloud services while decreasing the required cost and power consumption. Through testing of the speed and accuracy of recognition on the developed prototype architecture, I proved that the cloud platform architecture is a feasible solution for 3D object recognition.

Keywords: 3D object recognition, cloud computing, cloud service, machine vision, template alignment
# Table of Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Title Page</td>
<td>i</td>
</tr>
<tr>
<td></td>
<td>Acknowledgment</td>
<td>ii</td>
</tr>
<tr>
<td></td>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td></td>
<td>Table of Contents</td>
<td>iv</td>
</tr>
<tr>
<td></td>
<td>List of Figures</td>
<td>v</td>
</tr>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.1 Rational</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.2 Problem Statement</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.3 Objectives</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1.4 Contributions</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1.5 Limitations and Scope</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1.6 Proposal Outline</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Literature Review</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.1 Noise Removal</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.2 Fast Feature Point Histrogram</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.3 2D Interest Point Detection</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>2.4 3D Interest Point Detection</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.5 Recognition</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2.6 Cloud Computing</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>2.7 ROS (Robot Operating System)</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>2.8 Kinect Camera</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>Methodology</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>3.1 Cloud platform</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>3.2 Cloud-based object detection and recognition</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>3.3 Identification of specific objects</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>Experiment Result</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>4.1 Cloud</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>4.2 Recognition templates</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>4.3 Accuracy and Speed of Recognition</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>4.4 Objects recognition accuracy</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>Conclusion</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>5.1 Conclusion</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>5.2 Recommendation for Future Work</td>
<td>57</td>
</tr>
<tr>
<td>6</td>
<td>References</td>
<td>58</td>
</tr>
</tbody>
</table>
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Statistical Outlier Removal</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>FPFH</td>
<td>5</td>
</tr>
<tr>
<td>2.3</td>
<td>Narf Descriptor</td>
<td>6</td>
</tr>
<tr>
<td>2.4</td>
<td>SURE Descriptor</td>
<td>10</td>
</tr>
<tr>
<td>3.1</td>
<td>Cloud robotics platform based on ROS and OpenStack.</td>
<td>12</td>
</tr>
<tr>
<td>3.2</td>
<td>Recognition Method</td>
<td>13</td>
</tr>
<tr>
<td>3.3</td>
<td>Interaction Diagram</td>
<td>13</td>
</tr>
<tr>
<td>3.4</td>
<td>Interface</td>
<td>14</td>
</tr>
<tr>
<td>3.5</td>
<td>Template Alignment Example</td>
<td>15</td>
</tr>
<tr>
<td>3.6</td>
<td>Example of template matching.</td>
<td>15</td>
</tr>
<tr>
<td>4.1</td>
<td>Physical hardware rack at Rangsit University</td>
<td>17</td>
</tr>
<tr>
<td>4.2</td>
<td>Objects example</td>
<td>18</td>
</tr>
<tr>
<td>4.3</td>
<td>Can Template Matching</td>
<td>19</td>
</tr>
<tr>
<td>4.4</td>
<td>Template Sample</td>
<td>20</td>
</tr>
<tr>
<td>4.5</td>
<td>Object recognition accuracy in Experiment I.</td>
<td>20</td>
</tr>
<tr>
<td>4.6</td>
<td>Object recognition latency in Experiment II.</td>
<td>20</td>
</tr>
<tr>
<td>4.7</td>
<td>Red box accuracy</td>
<td>54</td>
</tr>
<tr>
<td>4.8</td>
<td>Red box matching scene</td>
<td>55</td>
</tr>
<tr>
<td>4.9</td>
<td>Powerbank accuracy</td>
<td>55</td>
</tr>
<tr>
<td>4.10</td>
<td>Powerbank matching scene</td>
<td>56</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Rational

Compute-intensive perceptual reasoning tasks such as object detection and recognition are basic behaviors that service robots (Cao et al., 1995) must perform in order to support and assist humans over a broad variety of different day-to-day activities. Effective object detection and recognition requires a great deal of data storage and compute power, more than would be typical of an embedded robot control system. If it were possible for the embedded system to offload the necessary storage and compute capabilities to a more cost effective centralized infrastructure, it may be possible to introduce service robots with vastly improved perceptual reasoning capabilities at very low costs.

This idea has led us to consider the applicability of cloud computing (Mell & Grance, 2011; Armbrust et al., 2009) to support service robots, especially in perceptual reasoning tasks such as object detection and recognition. There has been some research on solutions to open problems in the use of multiple networked robots, such as localization and mapping, cooperative robot learning, skills for service robots, and knowledge sharing. However, there is still a large gap between the abilities of individual or small groups of networked robots and what is required for truly useful multi-function autonomous service robots. Connecting multiple robots with a network helps us to share and pool resources, but architectures for Internet-scale distributed robot learning have not been extensively explored.

This paper presents a case study on the use of a cloud computing platform to support robotics applications, allowing robots to offload heavy compute tasks such as machine vision to cloud infrastructure.

1.2 Problem Statement

There are various ways for robotic machines to perform 3D object recognition. Template alignment is one of the basic method. Template alignment requires high fidelity templates to enable useful result. However, increasing the number of templates for object recognition increases the time required for a single machine to perform the recognition process. To solve the problem of time consumption, the most straightforward method parallel execution. But parallel compute hardware on the robot would come at the great cost. To avoid the time and/or cost required by dedicated hardware, cloud service can be introduced as the most efficient way to assign multiple machines to work on the alignment problem in parallel. Although recinizing more objects means using more storage, cloud infrastructure can help alleviate storage cost through sharing as well. The storage allocated can be changed based on the storage requirements for a particular project.
1.3 Objectives

In thesis, I aim to extend the capabilities of networked service robot systems, by:

1. Building a platform for experimentation on 3D object recognition.
2. Building a platform for allocating cloud resources to a robot.
3. Designing and implementing a cloud service using the platform mentioned above.
4. Deploying a 3D recognition algorithm to the cloud service.
5. Evaluating the prototype on the recognition of 100 common household objects.

I prove the concept of using cloud service, for the service robot application for the specific case study of 3D object detection and recognition. Experiments with the prototype demonstrate the feasibility of the use of cloud platforms to deliver improved perceptual, knowledge management, and reasoning capabilities of individual robots while keeping the cost and power consumption of the robot low.

1.4 Contributions

This thesis provides the groundwork for giving a widely distributed group of robots the ability to learn to detect and recognize 3D objects cooperatively. On the basis of this work, participating robots could share their knowledge with others via my cloud service. Although many researchers have pointed out the utility of cloud resources for robotic application, few have actually build up such a capability. This thesis is among the first research work to demonstrate the use of cloud resources in a practical service robot application.

1.5 Limitations and Scope

This project will use the following constraint:

1. The prototype is a proof of concept, not yet production-ready.

2. The point clouds used for recognition must be obtained from single Kinect camera; therefore, some objects may not be recognized accurately.

3. All of objects to be recognized need to be placed on a single plane or surface such as table.
1.6 Proposal Outline

Chapter 2 produces a review of literature on relevant algorithms, academic research, and open source software.

Chapter 3 presents my methodology, including the cloud platform, cloud-based object recognition, and specific object recognition.

Chapter 4 reports the experimental results from my work.

Chapter 5 concludes and gives a recommendations for future work.
Chapter 2

Literature Review

2.1 Noise Removal

One of the noise removal used in this thesis is statistical outlier removal. It correct the input point clouds by computing the mean \( \mu \) and standard deviation \( \sigma \) of nearest neighbor distances, and trimming the points that fall outside a nearest neighbor distance of the \( \mu \pm \alpha \cdot \sigma \), where the value of \( \alpha \) depends on the size of the analyzed neighborhood, as shown in Figure 2.1. For each point, I compute the mean distance from it to its neighbors. By assuming that the resulted distribution is Gaussian with a mean and a standard deviation, all points whose mean distances are outside an interval defined by the global distances mean and standard deviation can be considered as outliers and trimmed from the dataset.

2.2 Fast Feature Point Histogram

The Fast Point Feature Histogram (FPFH) by R. B. Rusu et al. (2009) is used to compute for a for an input point cloud with surface normals by following method:

\[
\text{FPFH}(p_q) = \text{SPFH}(p_q) + \frac{1}{k} \sum_{k=1}^{k} \frac{1}{\omega_k} \cdot \text{SPFH}(p_k)
\]

where the weight \( \omega_k \) represents a distance between the query point \( p_q \) and a neighbor point \( p_k \) in some given metric space as shown in Figure 2.2.

2.3 2D Interest Point Detection

Feature detection and description has been a main focus of research for many years. Detectors in intensity images has been an active focus of research. Feature detectors help find interest point in an image or object in an image. Interest point detection algorithms are designed to be invariant against moderate scale and viewpoint changes.

SIFT, by Lowe (2004), has proven that it provide robust matching across a substantial range of affine distortion, addition of noise, changes in 3D viewpoint, and changes in illumination. SIFT features found to be invariant to images scale and rotation. Detected features can be matched against a large database of features from many images are said to be very are highly distinctive correctly.

SURF, by Bay et al. (2008), can work as well as or even better than previous schemes in the terms of
Figure 2.1: Effect of sparse outlier analysis and removal: the original dataset is shown on the left, while the resultant one on the right. The graphic shows the mean k-nearest neighbor distances in a point neighborhood before and after filtering. R. B. Rusu et al. (2008)

Figure 2.2: FPFs. Red lines show the pair of itself and its neighbor. Some of the value which marked with thicker lines. Reproduce from R. B. Rusu et al. (2009).

distinctiveness, reputability, and robustness. SURE feature are able to computed much faster than the SIPT features. This outstanding performance is achieved by simplifying the processing and exploiting integral images in order to improve existing detectors.

2.4 3D Interest Point Detection

In decade years, 3D recognition is becoming popular in recognition research. Therefore, the 3D interest point detection is very important topic to discuss. Two of the very interesting technique are surface entropy for distinctive 3D features (SURE) by Fiolka et al. (2012) and normal aligned radial feature (NARF) by Steder et al. (2010).

NARF has designed for feature extraction that considers borders of object in the range image as a main feature. Its work can be divided it to 3 step: first is border extraction, then interest points extraction, and last is point descriptors. In interest points extraction NARF takes surface into account also. Last, importantly, the NARF descriptor can be calculated by using patch that has a size of $10 \times 10$ pixels. After that overlay a star shaped pattern onto this patch with focusing for change in the set of cells as Figure 2.4 shows.
SURE proposes to select distinctive points on surfaces using an entropy-based interest operator. The main features that are taken into consideration are changes on the surface and texture. To detect an interest point, local center of surface entropy mass is computed in a region of the local surface around a sample point. Its has 3 descriptor: for shape, color, and luminance. These descriptor are showed as histogram in Figure 2.4. Luminance is used along with color because color alone cannot detect differences between black and white.

2.5 Recognition

Object detection is a computer technology that related to the computer vision and image processing. Some of the example that include object detection are face detection and pedestrian detection, image retrieval and video surveillance. There are several technique that can be use for this purpose such as correspondence, spin image, and alignment.

2.5.1 Correspondence

A new approach to 3D object’s feature points correspondence is presented by H.-Y. Zhang et al. (2003). It has 3 steps. First, we evaluate the pose by matching the input image with reference image
using the aspect graph. Next, polar-exponential grid sampling and polar-log coordinate map are used to approximate feature point location of the moving target. Then the correspondence can be obtain by getting the means. This approach solve the problem of correspondence on plane object moving.

H. Zhang et al. (2010) present an improved 3D object feature point correspondence algorithm. Its methods can be classified into 3 steps. First, an object model database that based on aspect graphics is builted by an ART neural network, and quick search object model database algorithm is then proposed. The second and third steps are same as H.-Y. Zhang et al. (2003) method. This method solves the problem object model database having so many images and hard for searching.

2.5.2 Spin image

Spin matching method is capable of detecting object in a 3D reconstruction from photos. In addition, this technique allows retrieval of the orientations query of the model and uses residual error enables to distinguish between object with similar shape and a different shape.

Johnson and Hebert (1999) present a 3D shape-based object recognition system for simultaneous recognition of multiple objects in scenes containing clutter and occlusion. Recognition is based on matching surfaces by matching points using the spin image representation. The spin image is a data level shape descriptor that is used to match surfaces represented as surface meshes.

Halma et al. (2010) present a method for detecting object in unstructured 3D point clouds acquired from photos. First, the process require single spin image matching to find the location of the target area. Second, it uses the iterative closest point (ICP) algorithm to retrieve the quality of the match. Unlike the older version of spin images, no vertex surface normals needed, but a global orientation of the scene is used.

2.5.3 Alignment

Alignment method for recognition is introduce in recognition field with the propose of trying to align 3D object to solve the orientation.

Russ et al. (2006) presents a 3D approach for recognizing faces based on Principal Component Analysis (PCA). This work was applied with face recognition. It avoids elimination of size information by scaling a 3D reference to enable alignment of facial points important for PCA training, synthesis, and recognition. This approach shows the improvement in the recognition accuracy and dimensional reduction.
2.6 Cloud Computing

Cloud computing is a style of computing that aims to solve the problem of restricted resources in network systems. Cloud computing allows for on-demand access to resources. Importantly, it can solve the problems of computation and storage through parallel computation. Clouds system allow people from different places to interact with the same system at the same time. As a result, it allows user to access the system from different places.

There are many explanations of what is cloud computing. Mell and Grance (2011) has characterize important aspects of the cloud computing. They divide the characteristic into 5 essential characteristics and 2 type of models, service and deployment model. The characteristics are dividend into on-demand broad network access, rapid elasticity, resource pooling, self-service, and measured service.

1. On-demand self-service: User can use the computation service automatically.
2. Broad network access: Available over the network where any client platform can access it.
3. Resource pooling: The resource such as storage, processing, memory, and network bandwidth can be pooled to and use by multiple clients.
4. Rapid elasticity: The system can be scaled quickly and properly.
5. Measured service: Cloud system metering ability allows automatic control and optimize the resources.

The two types of models are service as a model and deployment as a model. Service models are classified into usage types, software as a service (SaaS) and platform as a service (PaaS). Software as a service means the application runs on the cloud infrastructure. The application can be accessed through a client interface such as Web browser. On the other hand, platform as a service gives consumers ability to control the application and ease deployment to the cloud.

Arumugam et al. (2010) propose a software framework implemented with Hadoop and ROS that provides parallelism and scalability advantages using cloud computing for service robots in large environments. This framework environment is a Software as a Service (SaaS) model, which means robots can share this service. The cloud computing environment provides robots to access compute intensive capabilities.

Rocha et al. (2011) presents the REALabs that a cloud computing for supporting network robotics applications. The REALabs cloud implemented the VM management service for VirtualBox. The REALabs provides VM management services interface and allows users to manage of their own VMs. These platform as a service offer a set of services for robotics application such as access control, federated authentication, and resource protection.

Jordan et al. (2013) describes the rising prospects of cloud robotic application. According to Kehoe et al. (2013), Willow Garages PR2 performed a complex and difficult task with cloud based as following sequence: the robot taken the object with a 2D and 3D image, but only the 2D image is sent for processing to the object recognition server. If server can identify an object, then send 3D CAD
model back to the robot. The robot uses the measured 3D point cloud data with 3D CAD model to determines the appropriate grasp strategies and stores results in the cloud server for future uses.

Turnbull and Samanta (2013) propose of a small scale cloud robotics that can be tested by offloading the computational load to the cloud infrastructure. Their system consist of multiple vision acquisition that sensor providing image data to the cloud. The cloud computes the location and behavior of the robots by using a recognition based algorithm, and then choose the appropriate commands back to the robots.

Tenorth et al. (2013) present knowledge-enabled cloud robotics application using the ubiquitous network robot platform. This system designed for the exchange of knowledge between robots. They implemented knowledge representation by using the web ontology language in the ROBOEARTH system. The scenario is presented as an recommendation robots in the convenience store. The sensor equipped with laser scanners for tracking customers and RFID tag readers for detecting the objects that have been picked up.

2.7 ROS (Robot Operating System)

(Quigley et al., 2009) propose ROS (Robot Operating System), a collectio of device drivers, libraries, visualizers, message-passing, middle ware, and package management support software that provide tools and libraries for heterogeneous and large-scale service robots in order to help software developers create robot applications. ROS is licensed under the BSD open source license, and is free for commercial and research use.

2.8 Kinect Camera

Kinect (codenamed in development as Project Natal) is a motion sensing input device by Microsoft. ROS supports the Kinect through two drivers: “freenect stack” and “openni kinect.” Both drivers provide point clouds, but the underlying implementations are very different and offer different benefits.

Kinect Product Features

1. Array of 4 microphones supporting single speaker voice recognition
2. Depth camera with 640×480 pixel resolution @30FPS
3. Color VGA motion camera 640×480 pixel resolution @30FPS
4. Fully compatible with all Xbox 360 models
5. Revolutionary technology that includes body recognition
Figure 2.4: SURE has 3 descriptor. (a) Shape. (b) Color. (c) Luminance. These figures are reproduce from Fiolka et al. (2012).
Chapter 3

Methodology

In this chapter, I propose an architecture enabling large-scale distributed robot learning through cloud computing, with a specific case study of object detection and recognition for service robots. After providing an overview of the platform, I detail how the architecture supports cooperative learning and explain my object detection and recognition approach.

3.1 Cloud platform

Figure 3.1 gives a high-level overview of the platform. The architecture assumes a private cloud running the OpenStack (Sefraoui et al., 2012) cloud operating system. Although a very similar system could be constructed using a public cloud such as Amazon EC2 controlled through an interface such as Amazon Web Services, with a private cloud I have full control over the hardware, visualization, and software stacks. OpenStack is designed to manage and automate pools of compute resources and provides scalable object storage using clusters of standardized servers. This allows us to offer on-demand computing resources to robotics applications, by provisioning and managing a large network of specialized virtual machines.

The cloud controller node distributes computational tasks to multiple compute nodes running robot application services. Each compute node provides services to support management and automation of pools of computer resources via visualization technologies. I have developed virtual machine instances responsible for performing 3D object recognition. I deploy a highly scalable object template repository using the OpenStack object storage service, which provides support for storing and retrieving arbitrary data on the cloud platform. Robots can access the cloud-provided services over the Internet by using ROS (Quigley et al., 2009). Processes running on individual robots use ROS messages (Quigley et al., 2009) for communication, and specific message types are marshaled and forwarded for processing on the cloud platform using a ROS proxy process.

3.2 Cloud-based object detection and recognition

Here I provide a description of the cooperative object detection and recognition approach built on top of the previously-described cloud robotics platform. I assume for the time being that each robot participating in the distributed system has a single Kinect and a Wifi connection to the Internet. I use the point cloud library (PCL) (Rusu & Cousins, 2011) point cloud data structure to represent the structure and appearance of known objects. The computational task of object recognition is likewise based on PCL functions.

The method is divided into two phases: training and identifying specific objects. Figure 3.3 shows an interaction diagram for the learning approach. To train his or her robot, a user can create and upload
templates of an object to the system using a simple application implemented as a ROS package. Refer to Figure 3.4 for a screen shot. This application require username and password to check the IP of the ROS master to use.

To detect and recognize objects, the user requests a 3D point cloud for the current target scene, which is then sent to the cloud based service. The scene will first receive at the network node where it will check the ROS master IP from the interface. Once the network node know the IP of the connected ROS master, it receive the scene and summit it to correct VM.

After ROS master node on the server-side, the virtual machine, receives the target scene, it tries to find an alignment between the target and a template for the matching result. The system runs multiple template alignment processes in parallel to reduce the time required to find the best template. This allows the robot to determine the position and orientation of the object in a scene.

The process of object template matching needs to store large amounts of template data to identify objects. In case the robot cannot respond to a request of the user, for example when the object is
unknown, then the robot should interact with the user and the learning module, uploading the acquired template to the cloud. In addition, a robot may learn to improve its skill and try to respond to request of multiple users at the same time. Cooperative learning and skill sharing between robots is a simple matter of sharing templates for specific objects.

3.3 Identification of specific objects

A flow chart of the object detection and recognition method is shown in Figure 3.2. First of all, I acquire a a Kinect image to get a point cloud. I remove outlier points using the PCL statistical outliers removal algorithm.

Next, I detect and subtract any large planar surface in order to segment objects from the background.
Figure 3.4: Screenshot of a simple application for capturing and uploading object templates as well as requesting recognition of an object currently in the field of view.

(usually a table or other flat surface). Finally, I identify specific objects using template matching follow the approach of Rusu et al. (R. Rusu et al., 2007). For recognition using template matching, I computes the point cloud surface normals and store it in kd-tree for search.

Finally, I perform template alignment, also known as template matching/fitting/registration, using the newly acquired data. A template is a small group of pixels or points that show a known part of a larger object or scene. By aligning a template to a new image or point cloud, I can find the position and orientation of the object that the template represents. I start by defining a structure to store the alignment results. It contains a floating point value that give score as the fitness of the alignment (lower the fitness means a better match of the alignment and a transformation matrix that explain how template points should translated and rotated in order to get the best alignment with the points in the target cloud.)

For calculating the fitness score, I use the sample consensus initial alignment algorithm to get the alignments fitness score and final transformation matrix. First, I select the sample point where the pairwise distance is greater than minimum distance. For each of them, I find which histogram is similar to the sample histogram. Then compute the rigid transformation from the sample point and correspondences. Then I compute the rigid error matrix from the point cloud. The rigid transformation can be decomposed into a 3-dimensional translation vector \((t_x, t_y, t_z)\) and a \(3 \times 3\) rotation matrix \(R\) as follows:

\[
T = \begin{bmatrix}
R & t_x \\
0 & 1 \\
0 & 0 \\
\end{bmatrix}
\]
Figure 3.5: Example of templates matching result for face.

Figure 3.6: Example of template matching.

Figure 3.5 gives an example of the result using template alignment.

In 3D object detection, I take a depth image containing an object and try to fit some previously captured templates to the new object. This works well for the position and orientation of the object in a cluttered scene (Figure 3.6), but finding the best match in the point cloud representation is computationally expensive, because the number of templates required is quite large. I therefore use cloud-based parallel computing to reduce the time required to compute the best match.
Chapter 4

Experiment Result

4.1 Cloud

We configured an experimental hardware setup modeling a complete cloud installation according to the topology shown in Figure 4.1. In this topology, the controller node, the network node, and the compute nodes are on separate physical cores. The servers are housed at at Rangsit University. This system is currently used for research only. We use one cloud controller node that can distribute computational tasks to the four compute nodes running the object detection application, and one network node to manage message sending, receiving, and forwarding.

The hardware and network are configured as follows:

- 2 rack-mounted server machines (Network node, Cloud controller node)
  - 2 Ethernet ports
    - eth1 - 1 real ip (110.164.186.47), eth2 - DHCP server (192.168.42.1)
    - eth1 - 1 real ip (110.164.186.45), eth2 - DHCP server (192.168.42.166)
- 4 PCs (compute nodes)
  - eth1 - connected to switch (192.168.42.11-14)
- 1 switch - 192.168.42.2

4.2 Recognition templates

We first created an initial sample template repository for the experiments containing 3D images in PCL format of a set of household objects. We implemented a user-facing template data collection task in order to enable easy collection of specific object templates and submission of those templates for storage in the cloud-based repository. Figure 4.2 shows examples of the objects. Figure 4.3 shows the examples of soda can template matching using the templates show in Figure 4.4.

We then performed two experiments to evaluate the accuracy and speed of identification of objects in the data set based on novel views not present in the original repository.
4.3 Accuracy and Speed of Recognition

In Experiment I, I evaluated the effect of the number of templates and number of virtual machines on accuracy. As shown in Figure 4.5, as one would expect, accuracy increases with the number of templates in the gallery but is unaffected by the number of virtual machines used. In Experiment II, we evaluated the effect of the number of templates and number of virtual machines on latency. Latency is calculated by measuring the time from the beginning of the submission of the request until we get the result. As shown in Figure 4.6, increasing the number of templates also increases latency but this effect can be mitigated by adding additional virtual machines. In practice, an effective balance between accuracy, latency, and resource utilization should be found.
Figure 4.2: Example household objects used in object detection experiments.
Figure 4.3: Template alignment for soda can templates.
Figure 4.4: Example templates for the soda can experiments.

Figure 4.5: Object recognition accuracy in Experiment I.

Figure 4.6: Object recognition latency in Experiment II.
4.4 Objects recognition accuracy

The following figures show the recognition accuracy for each object using the local system only. For each object, I used 10 times, each time in a random orientation in a clustered scene with two other objects. The two other objects did not have a similar shape to the target object. The distractions were otherwise randomly selected. The results are shown below.
From 100 objects, I chose 2 objects, a best case and the worst case, for further inspection. The best matched object is a red box, and the worst matched object is a small powerbank battery.

The red box recognition is 100 percent accurate because it is a big object. I could get detailed point clouds and it can only be positioned in few orientations. In addition, the two other objects have very different shapes, so the template will never match those objects. The object accuracy is shown in Figure 4.7, and the template matching result is shown in Figure 4.8.

On the other hand, the powerbank is a small object, I could only obtain coarse point clouds for creating templates. Moreover, other objects are larger than the powerbank, so the template of the powerbank can partically match those objects. The object accuracy is shown in Figure 4.9, and the template matching is shown in Figure 4.10.

![Figure 4.7: Red box template matching accuracy.](image)
Figure 4.8: Red box template matching scene.

Figure 4.9: Powerbank template matching accuracy.
Figure 4.10: Powerbank template matching scene.
Chapter 5

Conclusion

5.1 Conclusion

In this thesis, I have presented a combination of PaaS and SaaS service models for cloud computing applied to recognition of specific 3D objects by a possibly large and distributed group of service robots. Experiments demonstrate the feasibility of the concept. Cloud-based parallel computation dramatically reduces the time required to compute the best match between a gallery of templates and a given scene, and the orientation and position of the object in the scene can be retrieved.

The main practical limitations of the approach are that it does not perform well in identifying specific objects with similar shapes, and recognition latency needs to be improved. Using 3D key points and adding texture appearance modeling would be two ways to improve the identification of specific objects.

I find that the Kinect camera can always acquire good template point clouds. In particular, the Kinect does not detect small objects or transparent objects.

Moreover, communication is restrained by the ROS proxy. It adds overhead to communication between machines. Additionally, the Rangsit University network only allows connection on a few port for outside network communication. Which results in limitations to the connection services we can create.

Lastly, the cloud system is based on Openstack. Openstack is still in active development, so researchers have to keep up with updates. Not keeping up with updates may result in unnecessary work in debugging the software.

5.2 Recommendation for Future Work

This paper has presented a case study on the use of cloud computing infrastructure to support robotic applications, allowing robots to offload heavy compute tasks such as machine vision to cloud infrastructure. I recommend experiments lightweight methods of communication and other models for parallel computation such as map-reduce.

In the future, further increasing the number of templates storage would be an interesting experiment. We may find that increasing the number of templates results in higher accuracy. Doing so, we should be able to find the number of templates that giving out the best result and the highest accuracy. Also, further research should consider reconstructing the templates into CAD model instead of the using many templates of the same object from different perspectives. In addition, adding 3D or 2D key point based methods for recognition may give better results than template alignment in the terms of accuracy and speed efficiency.
References


