

**Nagoya Institute of Technology**

**Sato Laboratory**

2011 - 2012

**3D Natural User Interface using Kinect**

**LUCCIN Karim – LUONG Kevin**

Table of contents

[I. Acknowledgements 3](#_Toc318805144)

[II. Introduction 4](#_Toc318805145)

[III. Nagoya Institute of Technology 5](#_Toc318805146)

[1. Nagoya 5](#_Toc318805147)

[2. NIT presentation 6](#_Toc318805148)

[3. Sato Laboratory 7](#_Toc318805149)

[IV. Motivations & Background 9](#_Toc318805150)

[1. Kinect Project 9](#_Toc318805151)

[2. Background on Kinect 9](#_Toc318805152)

[V. Objectives & Realizations 10](#_Toc318805153)

[1. Objectives 10](#_Toc318805154)

[2. Used Methodology, Technologies and Tools 10](#_Toc318805155)

[a. Development method 10](#_Toc318805156)

[b. GANTT chart 12](#_Toc318805157)

[c. Control 13](#_Toc318805158)

[3. Solution 13](#_Toc318805159)

[a. Description 13](#_Toc318805160)

[b. Possibilities offered by the Kinect 15](#_Toc318805161)

[c. Limitations of the Kinect 16](#_Toc318805162)

[4. System Architecture 17](#_Toc318805163)

[5. Theory 18](#_Toc318805164)

[6. Technical 18](#_Toc318805165)

[7. Results 18](#_Toc318805166)

[VI. Evaluation of the project 19](#_Toc318805167)

[VII. Conclusion 20](#_Toc318805168)

[VIII. Glossary 21](#_Toc318805169)

[IX. References 22](#_Toc318805170)

[X. Appendix 23](#_Toc318805171)

# Acknowledgements

We would like to acknowledge the following people for their support and assistance with my internship.

The special thanks go to M. SATO Jun, our helpful supervisor. He guided and helped us with the various problems encountered through this project.

We would also like to thank the students of the Sato Laboratory who helped us greatly during our daily lives in Japan.

# Introduction

This project report is submitted for the M2 Student Exchange in Nagoya Institute of Technology, Japan.

We were given the opportunity to do a Research & Development project in Japan for 6 months. So we have decided to work with the Microsoft Kinect and have joined the Sato Laboratory.

In this report, we will first of all present the environment (Nagoya Institute of Technology and Sato Laboratory) and the motivations as to why we have decided to work on the Kinect before detailing the used methodology, works and results of the project. We will then evaluate our system and then discuss what we have learnt from this past 6 months.

# Nagoya Institute of Technology

## Nagoya

For this Exchange Student project we joined the Nagoya Institute of Technology (or NIT). As its name implies, this school is located in Nagoya: the 4th biggest town in Japan.

Located on the Pacific coast in the Chūbu region on central Honshu, it is the capital of Aichi Prefecture and is one of Japan's major ports. It is also the center of Japan's third largest metropolitan region, known as the Chūkyō Metropolitan Area.

Despite being one of Japan’s major ports, Nagoya has a large number of state and private colleges and universities such as the Nagoya University, Toyota Technological Institute and Nagoya Institute of Technology.



Image Nagoya

## NIT presentation



Image 2 Entrance of NIT

The Nagoya Institute of Technology is a public highest-level educational institution of science and technology located in Nagoya, Japan. Nagoya Institute of Technology was founded on 1905 as Nagoya Higher Technical School, renamed Nagoya College of Technology in 1944 then merged under the new educational system with the Aichi Prefectural College of Technology to be refounded as Nagoya Institute of Technology in 1949. Finally in 2004 it was refounded as Nagoya University Corporation Nagoya Institute of Technology.

This university holds many departments and laboratories:

*Faculty of Engineering*

* Life and Materials Engineering
* Environmental and Materials Engineering
* Mechanical Engineering
* Electrical and Electronic Engineering
* Computer Science
* Architecture and Design
* Civil Engineering and Systems Management

*Graduate School of Engineering*

* Materials Science and Engineering
* Engineering Physics, Electronics and Mechanics
* Computer Science and Engineering
* Architecture, Civil Engineering and Industrial Management Engineering
* Techno-Business Administration
* Frontier Materials
* Scientific and Engineering Simulation

*Educational Research Centers*

* Quality Innovation Techno-Center
* Ceramics Research Laboratory
* Research Center for Nano-Device and System
* International Center for Automotive Research

## Sato Laboratory

For this project we joined the Sato Laboratory. This laboratory was created in 1998 and is specialized in Computer Vision and Graphics: in particular 3D reconstruction from camera images, multiple view geometry (which is geometry composed of multiple cameras images).

The Sato Laboratory has worked on many kinds of projects:

* ITS (Intelligent Transport System)

The ITS is a system that makes vehicles “intelligent”:

With the help of cameras, the driver uses image information to assist drive and reduce accidents. Multiple cameras are put on the car to prevent dead angles.

This system can also create virtual images to prevent rain on the view mirrors and create a clear image.

* Human Computer Interaction

Using the Kinect they make applications where the user moves the cursor on the screen by moving the eyes.

* Mixed reality, augmented reality

Mixed reality is a 3D reconstruction of the scene using projectors.

La méthode au NIT c’est projeter des lumières codées sur des objets et le résultat c’est que les couleurs dépendent de la profondeur des objets.

* Research on special camera that can work on 4D information:

Image and light direction. Instead of having images in the form of I(x, y) we have I(x, y, θ, ω) where θ and ω are angles to the light sources.

This laboratory offers all the materials necessary for the students to do their research. A student room and a computer room are accessible by everybody any day/time.

# Motivations & Background

## Kinect Project

For this project we both instantly decided to work on the Kinect as we are interested in new kinds of experiences for the user; it can go from games such as we can see on Xbox 360 or devices like touch devices such as Microsoft Surface.

Another reason is that right now there is “boom” of touch devices. These devices are very popular such as smart phones or touch tables… Since the general public is keen on these devices it is the perfect moment to try and develop the next step to the touch user interface: the Natural User Interface or touch less devices.

## Background on Kinect

As I said earlier, we are right now in the “touch screen era”. Everybody has or has used touch devices such as smart phones, touch tablets… Companies produce more sophisticated touch devices.

People like to experience new ways to interact with computers or games. When the first “Kinect” project was revealed to the general public, which was called Project Natal at that time, people were amazed at the possibilities offered by this new gaming experience. That is why we turned our eyes to the Kinect as it was seen as the revolutionary gaming experience.

As of now the general public only sees the Kinect as a gaming device but more and more researchers do develop some new applications using the Kinect.

# Objectives & Realizations

## Objectives

We have set 2 objectives for this project:

* First we would like to see the various possibilities and limits offered by the Kinect and see how we can assemble the different devices to make a natural user interface.
* Secondly we would like to create some applications that use the potential of the Kinect such as a Web browser or a Media Center.

The goal of this project is to develop a natural user interface with Kinect. This means that the user will be able to interact with the computer via hand/arm motions. We will develop a web browser based on the design of Internet Explorer 10. This browser will be specifically adapted to Kinect using a Metro style. At the end the web browser will be able to understand the various input devices such as Kinect, the mouse and the keyboard.

## Used Methodology, Technologies and Tools

### Development method

The method used for this project is Agile. The main advantage of this method is to work by iterations. We first begin by working on the core of the system which has the least possibility of being modified through the project. At the end of iteration a meeting will be done to validate the solutions found.

The project is divided into 7 parts:

* Planning of the project

During this phase we first think of the functionalities offered by our system. By doing so we can then establish a Gantt chart and estimate the time needed for the development.

* Analysis

As it is a new technology to us we will have to study how the Kinect works and learn some new programming languages. During this phase we will also define the general and technical specifications that we will develop later.

The purpose of this phase is also to see the feasibility of the project by developing some prototypes or proof of concept.

* Conception

For this we will focus on the architecture of the system. From the prototypes done earlier we can see the different modules composing our system. We will use UML to draw the various diagrams such as class diagram, database diagram…

During this conception phase we will also sketch some rough screens of the user interface.

* Implementation

Now that we have decided the architecture of the system we will now implement the solution.

For the implementation phase it will be divided into 3 sprints:

* The first sprint will be on the data acquisition. By the end of this sprint we will have a functional module for gathering data from the input devices.
* The second sprint will focus on the algorithm of the movement recognition and some screens of the user interface. At the end of the sprint we will have some rough screens of the final user interface. By the end of this sprint each of us will work separately, one person will be in charge of the implementation of the movement recognition algorithm while the other will be working on the user interface.
* The last sprint will be the final version of our system. Everything should be implemented and functional at the end of the sprint.

### 

* Tests

For this phase we will test every functions of our system. The purpose is to make sure that there is no bugs anymore and that the project is fully working.

* Final check

In this last phase we check the documents that we have written for the project, clean the code, check that the application is fully functional…

### GANTT chartC:\Users\Kevin\Documents\Projet Japon\Gantt.png

Image GANTT chart

### Control

* Meetings

A weekly meeting with our teacher is done to see how the project is progressing. This meeting is also used to talk about the difficulties encountered in the project. During these meetings the teacher also indicates what could be improved on the current system and what would be a good idea to implement.

At the end of each phase/sprint, a little presentation is done to show what has been done so far.

* Bug tracking

Every bug encountered in this project will be written down as “not solved” with a small description of the bug and put in the Git versioning. Every time someone solves a bug, he has to push in Git the solution as well as a description on how he solved it.

## Solution

### Description

The project represents a Natural User Interface Web Browser using Microsoft Kinect. By using his hands as cursors the user could experience a new way to navigate on Internet.

But for this system to be fully enjoyable and useful for the user some requirements are necessary.

The system must be responsive; having time lags between each click would ruin the user’s experience. That is why we developed this project with the constraint of making the system run in Real Time (30 frames per second).

We originally planned to put the web browser into 2D Window System. But as it turned out quite bland, we have decided to switch to 3D Window System. Since the project is a Natural User Interface, it is very important to appeal to the user and having 3D Window System offers more possibilities in terms of User Interface than 2D Window System.

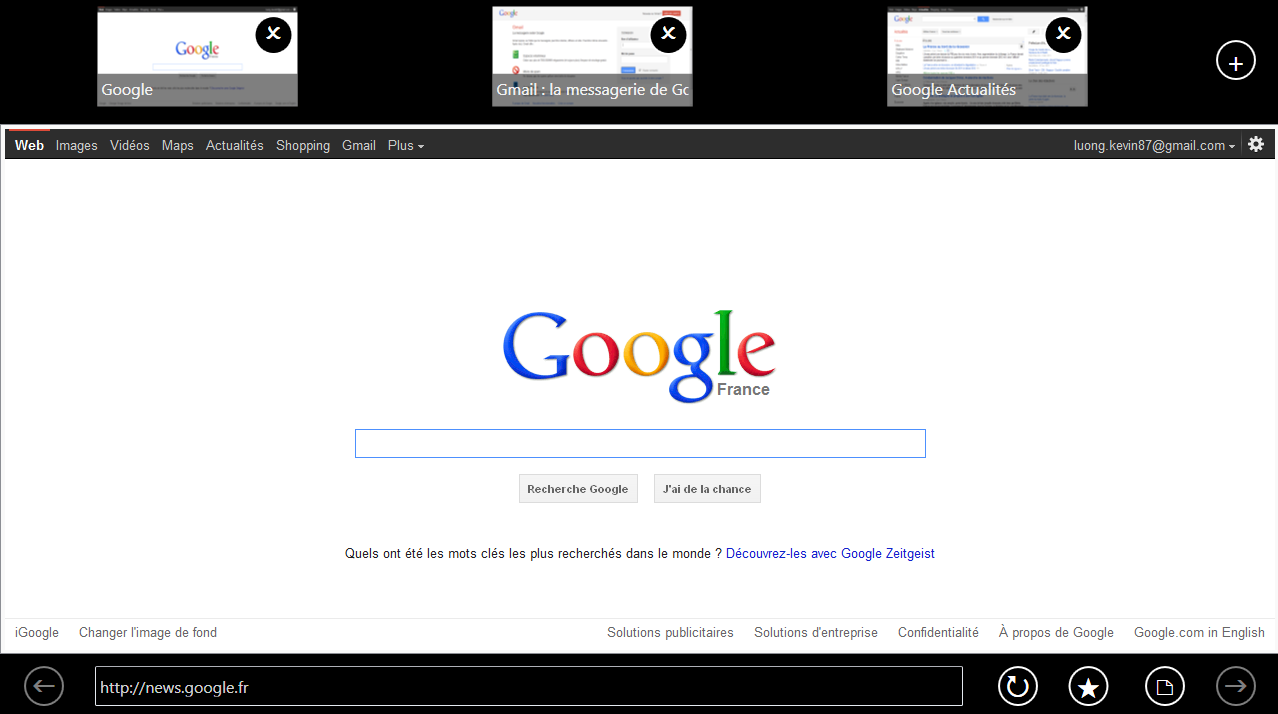


Image 4 2D Window System

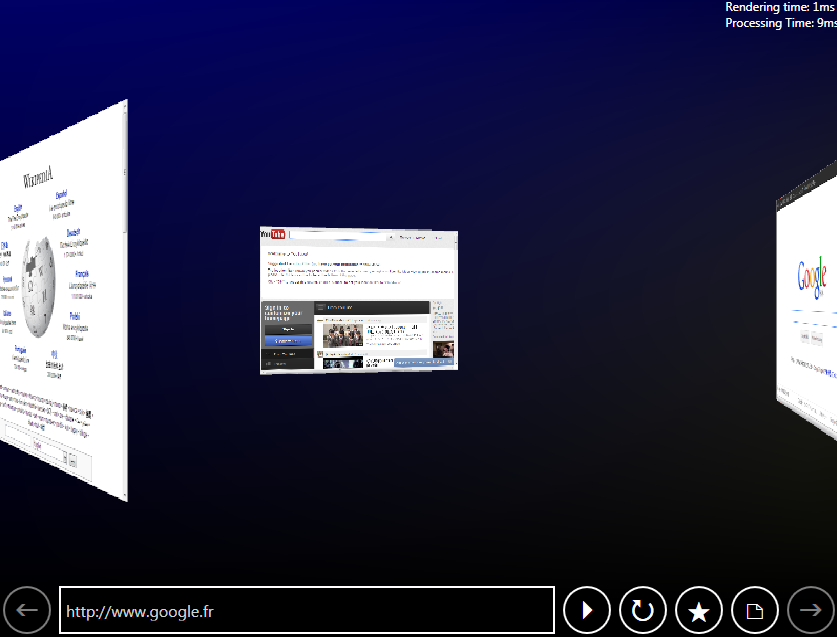


Image 5 3D Window System

### Possibilities offered by the Kinect

The Kinect is a motion sensing input device by Microsoft. Based on the design of a web cam, it enables the user to interact with applications using gestures or spoken commands. It includes 3D depth sensors, RGB camera and a multi-array microphone.

In 2011, Microsoft released a Software Development Kit (SDK) to allow developers to write Kinect Applications. This SDK allows us to access raw streams of data captured by the cameras of the Kinect in the form of 4 data features:

* Depth data
* Color data
* Skeleton recognition
* Voice recognition

### Limitations of the Kinect

During our first phase of our project we had to identify the various limitations that the Kinect has. From this phase we have identified 3 main limitations:

* Distance: Kinect only recognizes shapes and objects that are at least 1 meter away from the camera.
* Accuracy: Some noises appear on the data output which makes the accuracy a little instable.
* Recognition: you have to be standing up first for the Kinect to recognize your skeleton. When the user is sitting down the Kinect won’t be able to output your skeleton correctly.

Another problem with the recognition is the ambient lighting. Because the lighting differ from each place and moment the Color data given by the Kinect can be false. It might not recognize some elements because it is too bright or too dark.

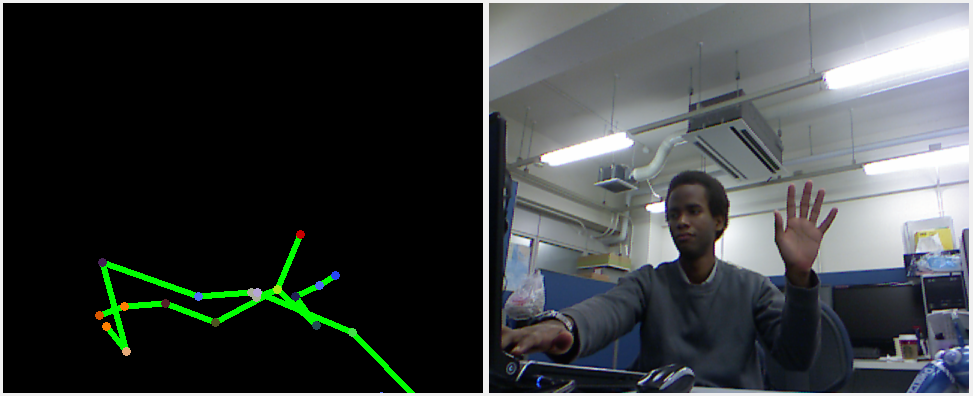


Image Skeleton recognition problem

## System Architecture

Image Class Diagram

### KinectBrowser

KinectBrowser is the name of our system. This part of the system is the main core: it initializes all the elements necessary for the system such as the Web Browser, the main window and the keyboard.

### Input.Kinect, Input.Mouse & Interaction

Since we wanted our project to both utilize the Kinect and the Mouse it is natural to have our system detect these two entry devices. As its name implies Input.Kinect gets raw data from the Kinect whereas Input.Mouse gets the data from the mouse device.

At this point Input.Kinect gives all the data: Depth, Color and Skeleton data. These data are then filtered and treated by the Interaction.dll and the ImageProcessing.dll. From the limitations we have found earlier we have decided to only use the Depth data.

The Interaction.dll is used to recognize the hands of the user as well as the gestures done by the user.

### ImageProcessing

ImageProcessing.dll is used to treat data from the Kinect on the graphical card. Since our objective is to make a system that runs in Real Time this processing on the graphical card is mandatory.

When the system gets the raw Depth data from the Kinect, we couldn’t determine what part of the data represents the hands of the user. ImageProcessing.dll tracks the regions (or blobs) on the screen and determines where the hands of the user are. This part will be explained in the “Technical” part of the report.

### D3D

D3D.dll or Direct3D.dll is part of Microsoft’s Application Programming Interface (API). It is used to render 3D graphics in applications, allow applications to run in full screen, apply shaders…

## Theory

### Algorithm to recognize hands

From what we have said earlier about the different features offered by the Kinect we have decided to only use the Depth data as it was reliable under any conditions (poorly light room, bright room, user sat down…).

Our system recognizes the hands of the user and uses them to control cursors. But having only the Depth data means we would have to execute an algorithm to recognize the hands and isolate them from the rest of the body.

Since the user needs to move his hands in order to control the cursors means that the closest parts of the body to the Kinect are the hands. They would typically be in front of the user as he is using our system. Applying this logic to the Depth data from the Kinect we would filter the pixels based on their distance from the sensor of the Kinect, the closest pixel from the sensor being the tip of the hand.

Once the pixels are filtered we would then group these pixels by blobs because the closest blobs are more likely to be the hands of the user.

When all the blobs are done the system would track only the 4 biggest blobs on the screen and label them as potential hands candidates. We have decided to track the 4 biggest blobs because usually the Kinect detects both hands and the head. The last blob is added for safety because sometimes there can be some noise from the Kinect’s sensor.

### CPU vs. GPU

The objective of our system is to run in Real Time. So during the Analysis phase of the project we have concluded some experiments on how to improve the speed processing of our system. One way to improve it was to execute the image processing directly into the graphic card instead of the CPU.

Used range

Image 8 Speed processing on the GPU and the CPU

As you can see on this comparison image between the GPU and the CPU there is a huge gap between the speed processing done on the GPU and the CPU.

The CPU tends to be faster as the image gets smaller whereas the GPU is constant and always very fast.

## Technical

## Results

# Evaluation of the project

# Conclusion

# Glossary

***SDK***: “Software Development Kit” set of programs and libraries to help develop applications.

***Kinect***: Microsoft camera used to capture gestures done by the user.

***User interface***: Visible part of the program for the user where he can interact with the application.

***Metro design***: Design for user interface which is appropriate for touch devices (large buttons, wide text…).

# References

# Appendix