3D DANCE HEAD USING KINECT AND 3D PROJECTOR

Supervised by Prof. LYU Rung Tsong Michael

Student: Chan Wai Yeung (1155005589)

Lai Tai Shing (1155005604)

- Introduction
- Objectives
- 1st term review
- 2nd term work
- Design & Implementation
- Limitation
- Conclusion
- Q & A

Introduction

- Participants' heads are superimposed to the dancers
- Limitation of current Dance Head

- Inspiration
  - Play Dance Heads without the limitation
  - Use the popular device Kinect to capture the head

Objectives

- Study the Kinect SDK or other libraries
- Design and implement an algorithm to extract the “Head” part using Kinect
- Rendering the image in 3D
- Study Point Cloud library used for reconstruct the 3D point image.
- Make some special features for the application

Development tools

- Programming Language (C++)
- Open source libraries
  - OpenNI
  - OpenCV
  - Point Cloud Library
- Operating system
  - Linux (1st term)
  - Windows (2nd term)

1st term work

- Study open source libraries
  - OpenNI
    - As a channel to get data from KINECT
      - Color image
      - Depth map
  - OpenCV
    - Cascade classifier for detection of face
    - Image processing

Different methods

- Cutting the head by radius
- Edge detection
- Surface normal
- Surface normal combined with depth value


DOGS
LIBRARY

Overall work

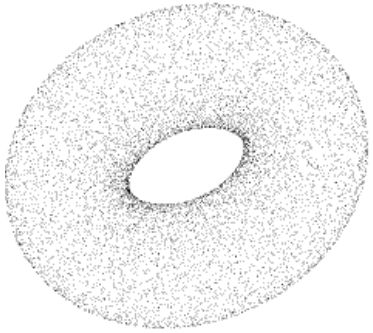
- Semester 1
  - Capture image from Kinect
  - Face Detection OpenCV
  - Extract people from background
  - Extract the head part by surface normal
- Semester 2
  - Pass image data to Point Cloud Library
  - Reconstruct the 3D point image
  - Add special features to applications

Optimize seml's work

- Multi-faces detected
  - Use extra variable to save characteristics of different face
- Improve frame rate
  - Reduce the computation of parts
- Speed up the face detection
  - Scale down the image resolution
- Improve the cutting edge of head
  - Estimate the edge values by the near points

Point Cloud Library

- Free and open source library
- Run on different platform
  - Windows/ MacOS/ Android/ IOS
- Data structure used to represents wide range of multi-dimensional points
- 3D point cloud
  - Represent X, Y, and Z geometric coordinates
- Use point to reconstruct the captured object



- Compatible to hardware sensors
  - Kinect
  - PrimeSensor 3D camera
- 2D / 3D image processing
- 9 modules
  - Filters / features / keypoints / registeration / kdtree / octree / segmentation / sample_consensus / surfae / range_image / io / visualization

Point Cloud data type

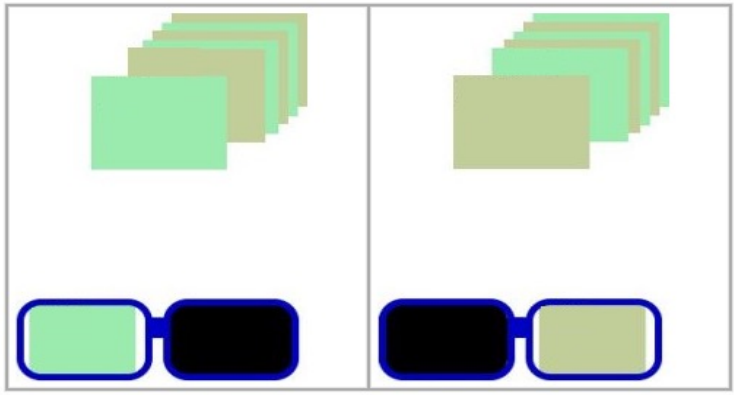
- Point
  - RGBXYZ
  - XYZ
  - RGBXYZA
- Point Cloud
  - Collection of points

3D stereoscopy

- With glasses
  - Active
  - Passive
- Without glasses
  - Time-multiplexed
  - Spatial-multiplexed

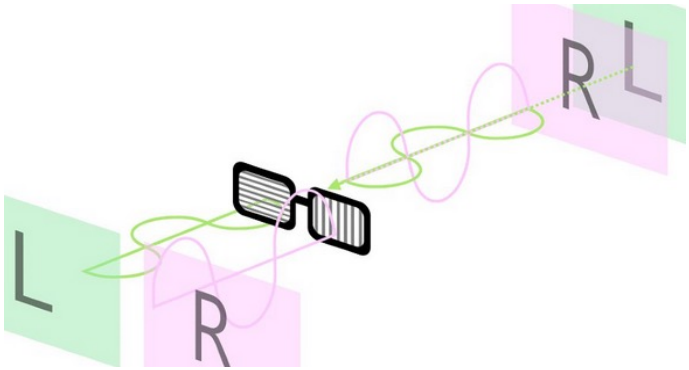
Active glasses

- Shutter glasses
- Screen would alternately display the left-eye and right-eye images
- Glasses would shield left and right alternately

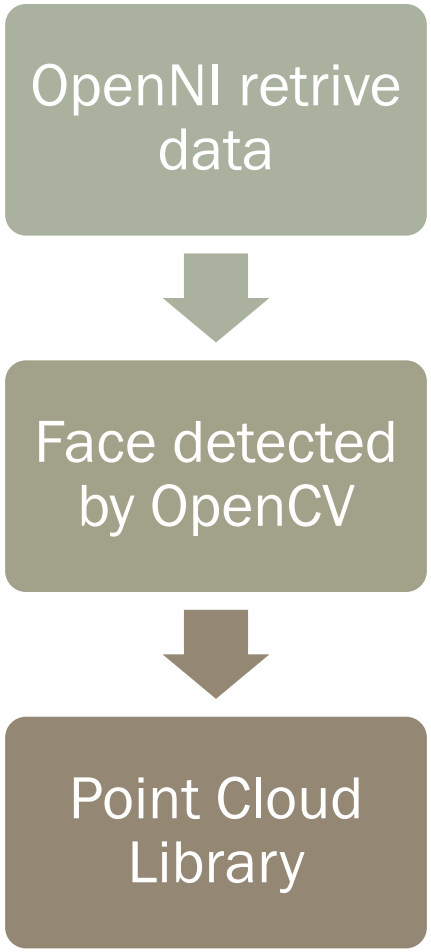


Passive glasses

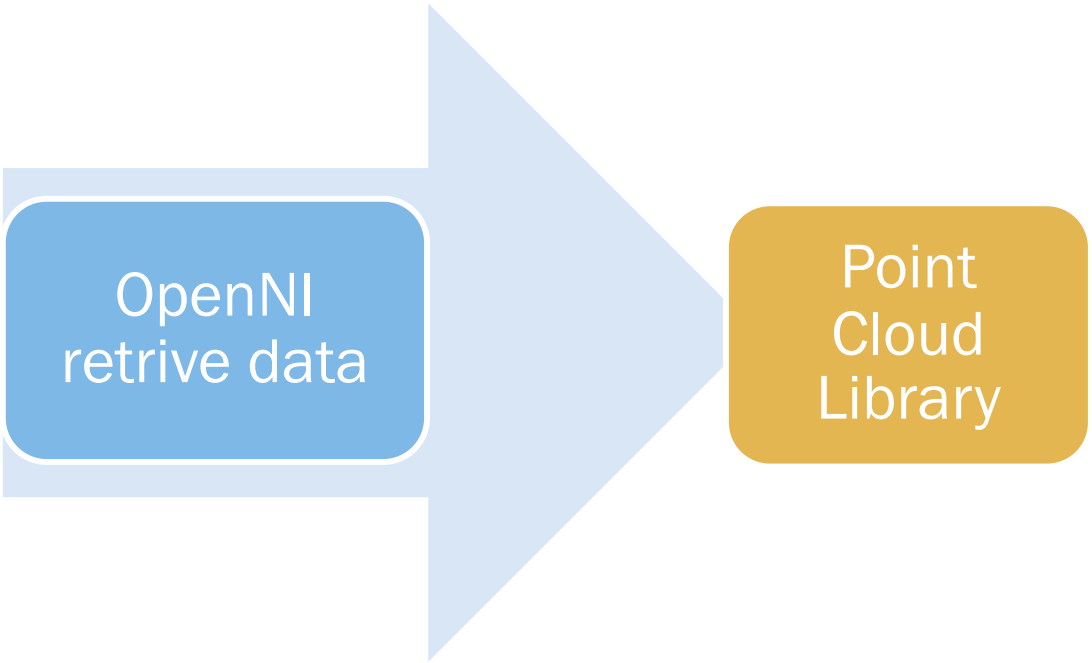
- Use special glass to filter out different image
  - Red-Blue
  - Polarizer



Design


OpenNI retrive data
Face detected by OpenCV
Point Cloud Library

- openni_wrapper



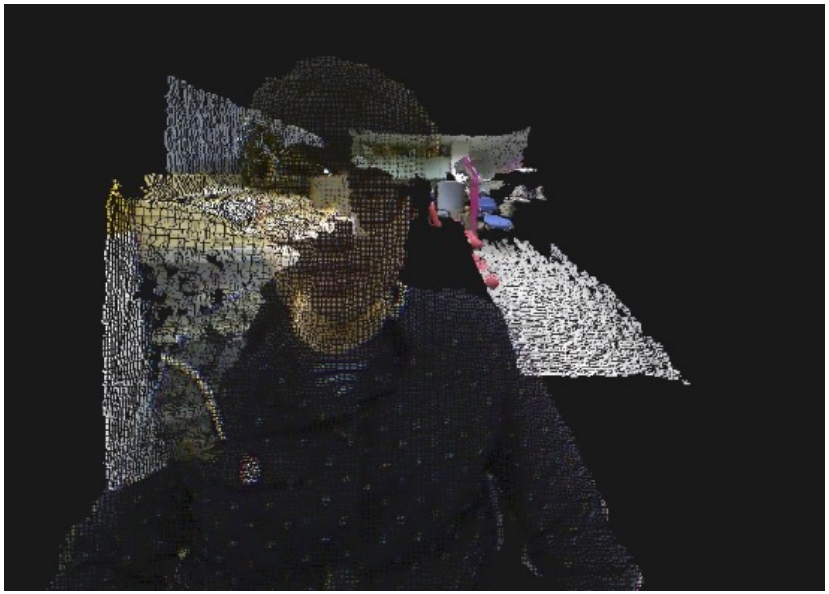
Implementation

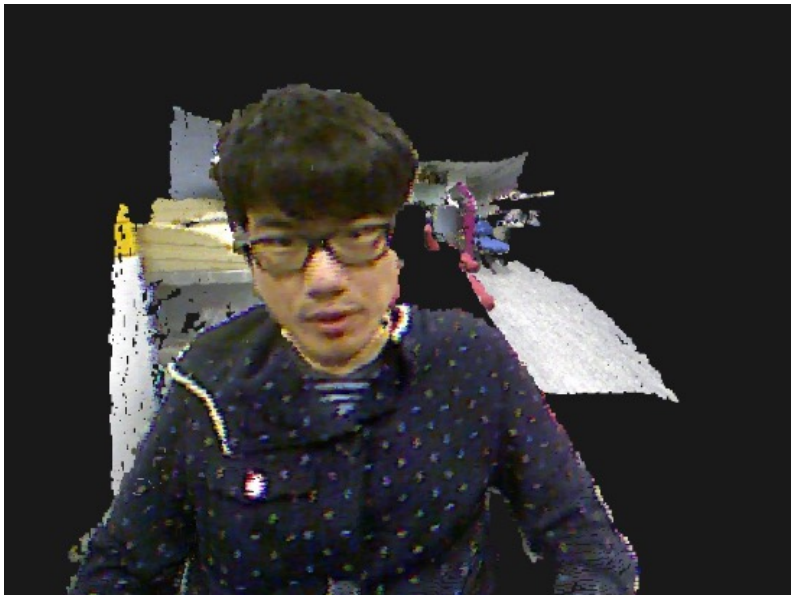
- Module io
  - Save and load background
- Module visualization
  - Render or draw 3D shapes

Features

- Blur function
- Swap faces between two players
- Enlarge or reduce the face size
- Load image to the background
- Move forward or backward to the image of the players

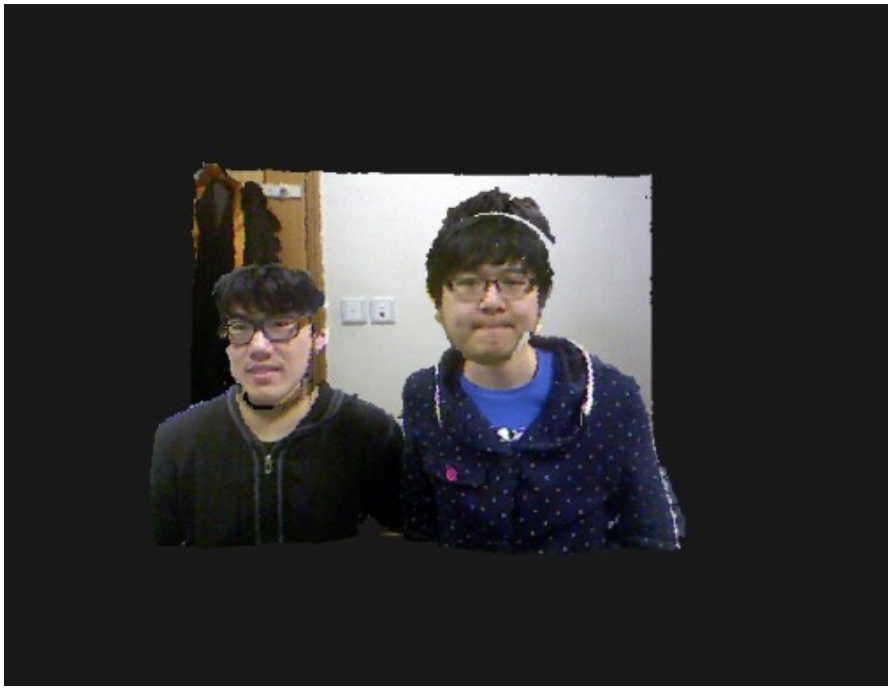
Blur function





To smoothen the surface of point cloud

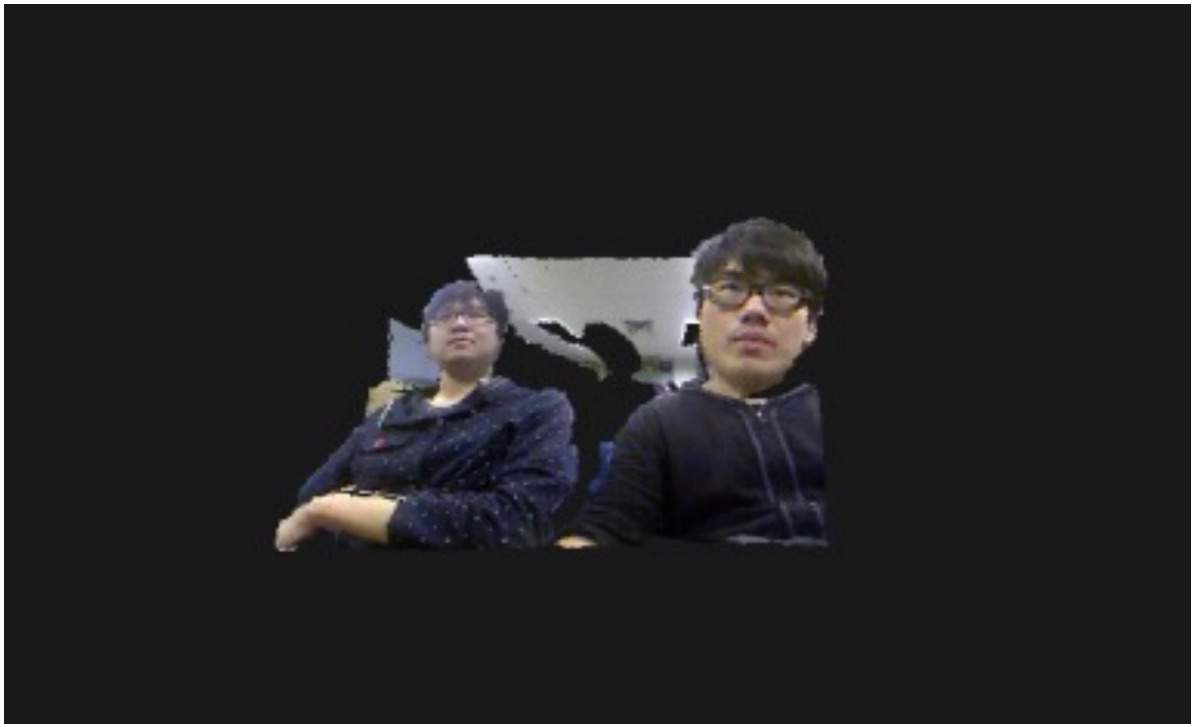
Swap faces between two players



Two face detected at the same time

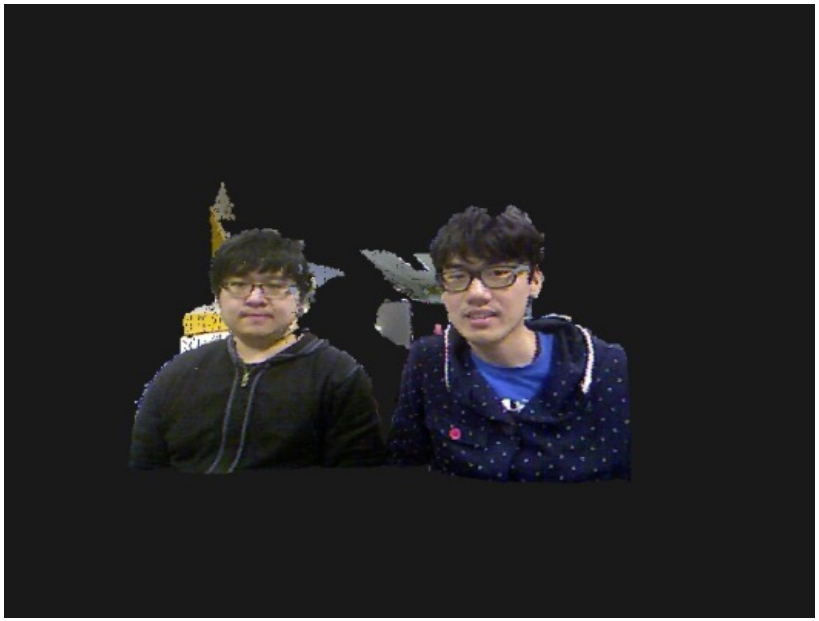
“v” : swap the faces between 2 players

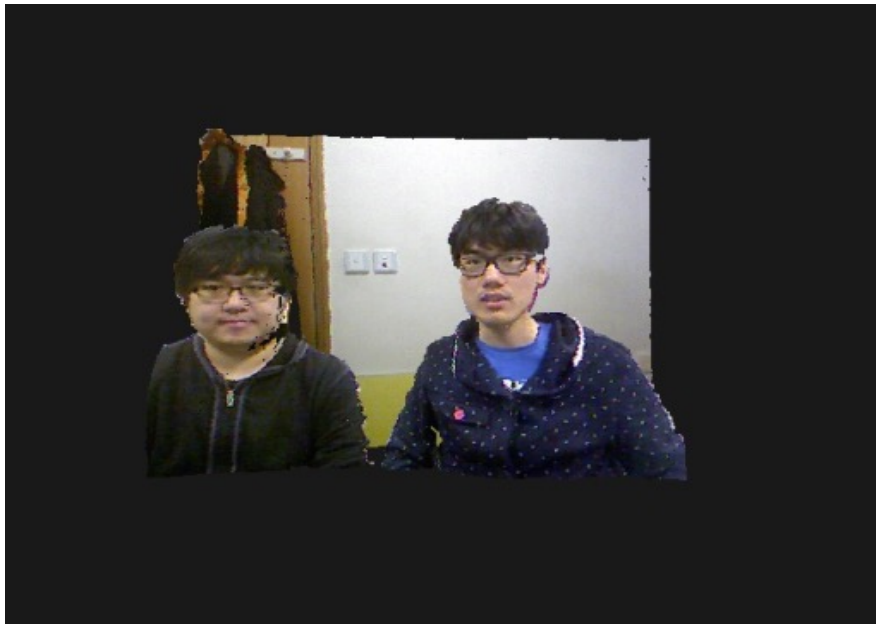
Enlarge or reduce the face size



“,” “.” : enlarge and reduce the size of the players’ faces

Load image to the background



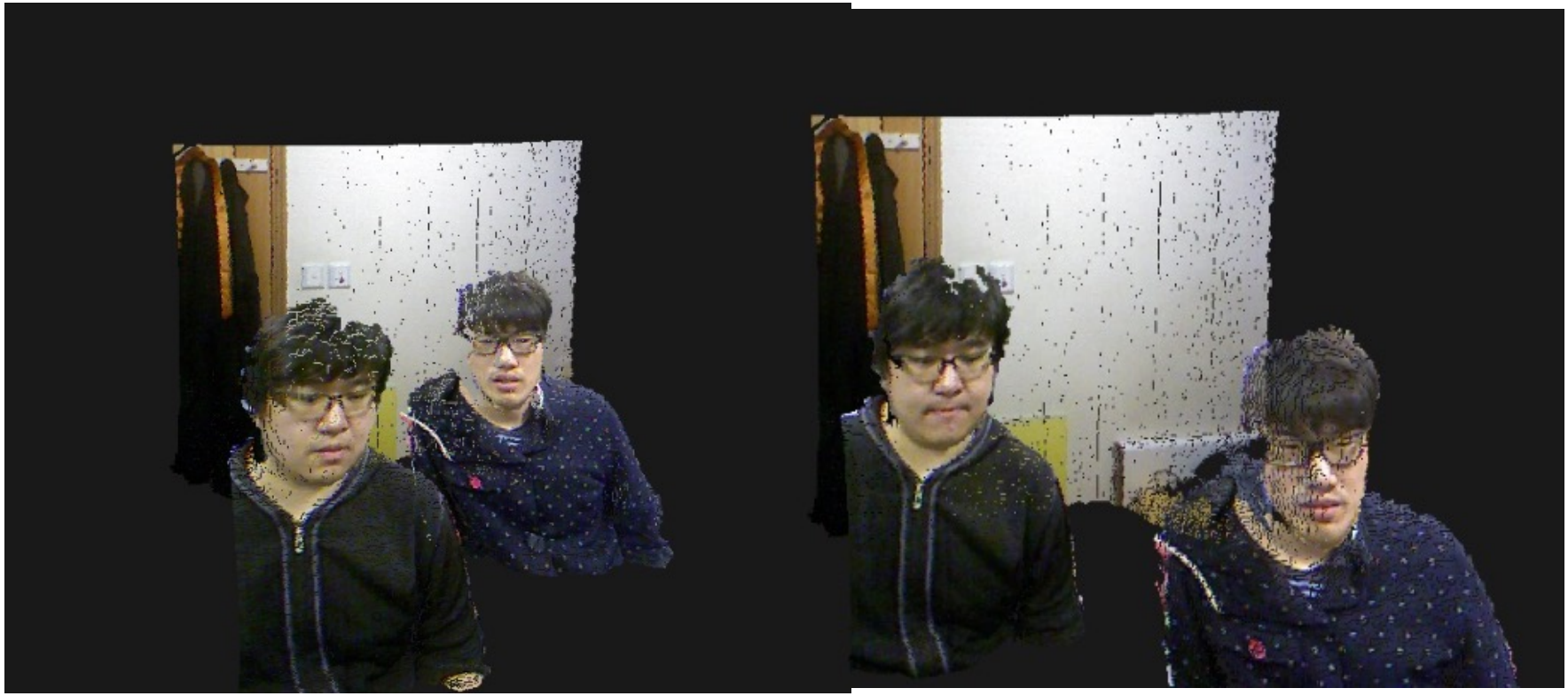


“k” : capture the image and save as background

“l” : load the image to the background

“b” : change or reset the background

Move forward or backward to the image of the players



“[” ”]” : move forward or backward of the players

3D display

A demo of red blue 3D stereo

Limitation

- Side view problem
  - Cannot detect side view of players
- Quality of the output image
- Kinect limitation
  - Errors occur when detecting black objects
  - Black colour absorb infra red
- Error of face detection
  - The cascade may not fit all kind of faces
- Low frame rate
  - Heavy pixel by pixel computation

Conclusion

- Image processing
- OpenCV
- PCL
- 3D point image
- Face detection

Q & A