

# Who are we?

Vibot and MAIA students:



Malav BATERIWALA



Eduardo OCHOA



Mohit VAISHNAV



Liliana VALENCIA

# *Why is this project important?*

- Improve the home-made acquisition and processing software of previous works
- Learn and apply different C++ tools
- Make research on different academic topics (as the Registration problem)
- Propose different approaches for a problem, as the OpenCV approach over last year used algorithms

# Background and Challenges

Main features of last year projects:

- Group 1
- Group 3

Challenges:

- Program and libraries installation
- Understanding their methods
- Propose new approaches
- Acquisition issues

# Project Management

## Task Distribution

## Project meetings

- Weekly meeting
- Project meeting

## Tools

## Success



Group Members	Leadership and Task Division
BATERIWALA Malav	Supervising overall Project Organization of code
VAISHNAV Mohit	Research Acquisition
OCHOA Eduardo	GUI Feature Matching
VALENCIA Liliana	Overall Management ICP



<https://github.com/bigmb/HakunaMatata.git>



<https://hakunamatataub.slack.com>



# Overlook

- Developed by Intel
- For both academic and commercial use
- Has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android
- Library of programming functions written in optimized C/C++, the library can take advantage of multi-core processing.
- Adopted all around the world



# Advantages over PCL

- Advantage over the feature detection algorithms proposed
- Possible to get the images from the Kinect and the using libraries and functions that work in image space.
- Better features for the registration process
- Improvements in speed in the algorithm and software
- Matching two frames of the body in image space
- Possible to make pre-processing in the data



# Data and the Kinect

- Microsoft API's implementation for the grabber
- Obtained Depth Image
- Obtained Color Image
- Acquisition of Kinect Coordinate system
- Possibility to have multiple captures

## KINECT MICROSOFT METHODS

ColorFrameSource

DepthFrameSource

CoordinateMapper





# Feature Matching

- Different kind of algorithms
- SIFT algorithm used
- Better features identified
- Matches according to the keypoints found between two images
- Robust match of the first eight best matches



## OpenCV SIFT Functionalities

sift.detect()

sift.compute()

sift.detectAndCompute()

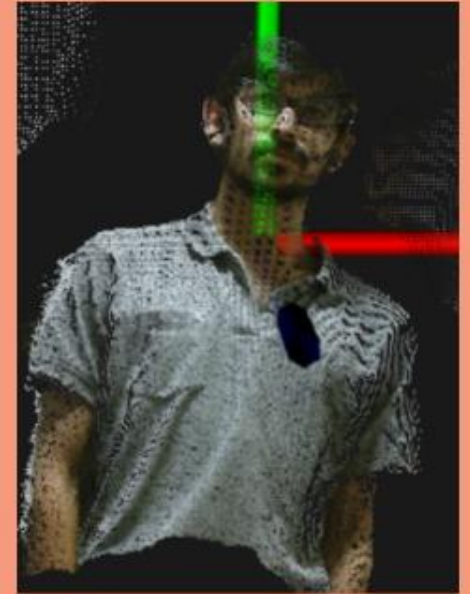
cv2.drawKeypoints()

Cv2.FlannBasedMatcher()



# Registering Point Cloud data

- Faster and more stable than Standard ICP
- Usage of ICP Normal, point to plane distance instead of Point to point
- Better results with initialized features
- Transformation matrix between images with OpenCV functions



*Improvements*

спасибо  
danke 謝謝  
ngiyabonga  
tesekkür ederim  
dank je  
gracias  
tapadh leat  
mochchakkeram  
go raibh maith agat  
arigatō  
takk  
dakujem  
merci  
ευχαριστώ  
terima kasih  
감사합니다  
sagolun  
sukriya  
kop khun krap  
grazie  
bedankt  
dziękuje  
obrigado  
hvala  
mauruuru