

EXTERNSHIP PROJECT

Title

Face Detection Robot Simulation Using ROS

Date: 2nd, August 2021

Team Members

Divyansh Agarwal

Dishika Poddar

Abin Gireesh

Rajat Srivastava

Introduction

a. Overview:

The complete application that was developed is a robot application in the simulation world integrated with vision and face detection capabilities. It will sense and detect the person's faces in the simulation environment and perform specified actions depending on our requirements. We will use the ROS Melodic version and python OpenCV to detect faces with Visual studio code as IDE to develop this simulation.

b. Purpose:

A face-based user verification robot serves many purposes in places like a multi-specialty hospital with tens of doctors and hundreds of patients. In places like that, directing the patients to the concerned doctor is a difficult and risky task for a human to do as the diseases are contagious. So, a face-based user verification robot checks for the scheduled appointment of the patient with the respective doctor and navigates them to the particular chamber without human intervention. This robot can be deployed in many places like offices, larger companies, and different other places where authentication and security are the utmost priority. In this project, we shall create one such robot application in simulation and perform the face-based user verification tasks using ROS and OpenCV

Literary Survey:

a. Existing Problem:

In places such as hospitals it is extremely risky to escort patients from one place to another since they could be contagious.

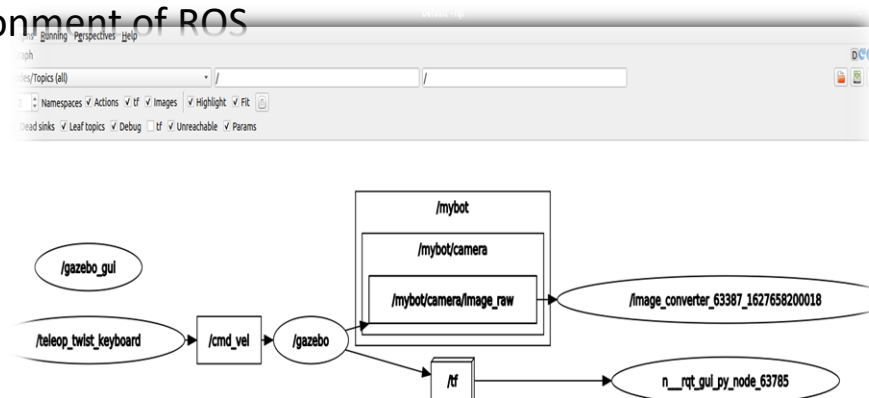
b. Proposed Solution:

A robot application integrated with face detection was built. It will sense and detect the person's faces in the simulation environment and perform specified actions depending on our requirements. We will use the ROS Melodic version and python OpenCV to detect faces with Visual studio code as IDE to develop this simulation.

Theoretical Analysis:

a. Block Diagram:

The robot is simulated in Gazebo environment, and we need to connect it for image transfer using ROS topics and concepts of Nodes. The robot is designed in Model editor and then brought to Gazebo real time. The data of images is converted from ROS to OpenCv using CVbridge. This provides the functionality to process images across both the platforms using Python Packages. Rest flow of data is brought from Gazebo to RViz, and tested in real time environment of ROS.

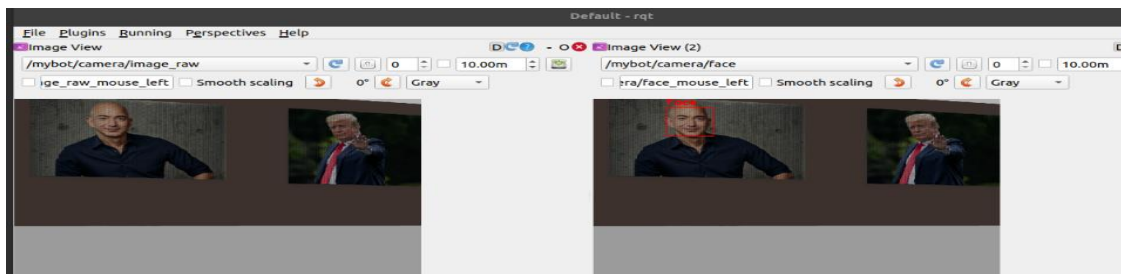
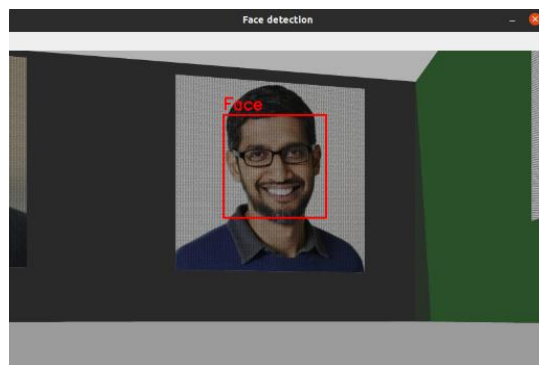
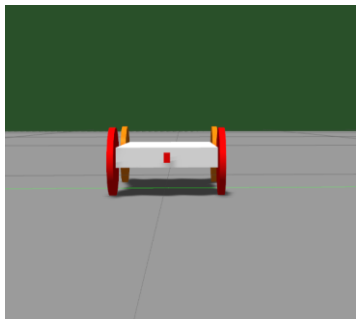


b. Hardware /Software Designing.

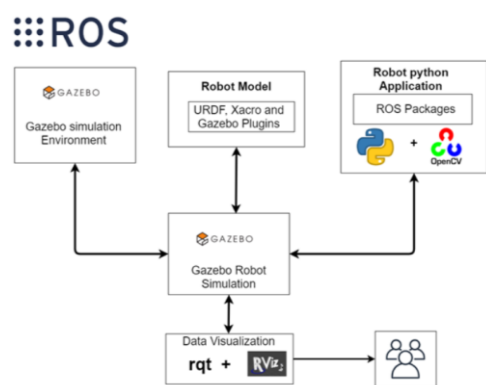


Experimental Investigations:

While running the files and experimentation, we learned a lot about ROS and packages. Also using the ros wiki website allowed us to overcome many errors as well help us understand various packages. Also making the python scripts we worked on OpenCV and learned it details. This experimentation helped us understand the complex structure of ROS and work with various packages and files like Package.xml and CMakeLists.txt



Flowchart:



RESULTS:

In this project we have successfully deployed Open CV and with the ROS packages we have developed a Face detecting robot which can detect faces, and wish to further explore and incorporate how "Computer Vision" can be incorporated in various other fields of Robotics.

Link to the Demo : <https://youtu.be/AgMBR2EbX k>

ADVANTAGES:

- ROS and OpenCV open sourced, therefore there is not redistribution problem or licensing issue when using for industrial or commercial or personal level projects.
- Prebuilt packages and Reusability reduces the time to rebuilt a real system from base , as ROS works as Meta Operating System.
- Low cost and Time saving
- Active community and documentation helps in learning , solving bugs and finding solutions easily.
- Regular updates , keeps the packages updated and fixed with previous big fixes

DISADVANTAGES:

- ROS is complex as , one needs to learn programming to understand the working
- Multiple distribution/versions , result in inter in-capability of system to use common packages easily , as per naming convention

- Not on major Operating Systems , as ROS2 is being built , it is still with many glitches
- Backward compatibility with older packages is not easy

APPLICATIONS

A face-based user verification robot serves many purposes in places like a multi-specialty hospital with tens of doctors and hundreds of patients. In places like that, directing the patients to the concerned doctor is a difficult and risky task for a human to do as the diseases are contagious. So, a face-based user verification robot checks for the scheduled appointment of the patient with the respective doctor and navigates them to the particular chamber without human intervention

Especially during this pandemic this bot will be extremely useful in reducing human contact.

It can be used in places such as banks and offices as well, to escort people while maintaining a safe distance from them.

It can be used at old age homes and schools, it reduces their exposure to human beings which is quite necessary since they are the most vulnerable population for the upcoming third wave of the novel coronavirus.

CONCLUSION:

We would like to thank our trainer Mr. Durgaprasad Bethi and Smart Internz, for the support and guidance through this entire training program that has helped us make this project.

The project can be deployed in places where Autonomous facial recognition could be used. This project is just one of the many

applications that can be developed using Open CV and ROS, this project can be further developed for more enhanced and better results with time and potentially be very valuable product.

FUTURE SCOPE

This bot can be made absolutely autonomous which would completely eradicate the need for human intervention. It can be integrated with a image recognition model that can identify humans who are unknown and have features such as buzzers or other forms of alarm and a web interface that warns the user of the trespasser on their property. Hence it can act as a huge asset in places that need high security while keeping humans out of the risk.

BIBLIOGRAPHY:

- 1) https://smartinternz.com/Student/guided_project_info/4807#
- 2) <https://github.com/Durgaprasad-SB/Face-Detection-Robot-using-ROS>

APPENDIX:

a. Source code

b. Robot output Screenshot.

Link to the Demo : https://youtu.be/AgMBR2EbX_k

Link to Source Code:

<https://www.dropbox.com/sh/kz1a5qanw6dkx20/AAAbxB4OpeJ5pWmmS7LmugOoa?dl=0>