ROBOGRAPHERS

FACIAL EXPRESSION RECOGNITION USING SWARMS

CRITICAL DESIGN REVIEW REPORT

SPONSOR: DR. KATIA SYCARA

TEAM G

JIMIT GANDHI

GAURI GANDHI

ROHIT DASHRATHI

SIDA WANG

TIFFANY MAY

ABSTRACT

This is a comprehensive report that reviews the design of a system of swarm mobile robots which would collaboratively analyze facial expressions and click pictures in a social occasion such as parties, weddings, pleasant ceremonies. The development of this project required us to enquire and discuss various requirements that would be critical to any usual social occasion use case. Once requirements were defined, a series of discussions and brainstorming lead to designing of functional and cyber-physical architecture involving various subsystems. This in turn led to development of a project schedule which lays down dates for completion of step-by-step progress and milestones thus fulfilling the requirements slowly. This report describes each of the above aspects in detail.

The preliminary execution of this schedule was carried out and results were recorded. Analyzing these results gave us a sufficient proof of the feasibility of the planning stage as well as brought to light various potential risks and problems, some of which were already taken care of. The entire report concludes with the lessons and takeaways from the experience till date and in addition also lists down next major steps towards completion.

Table of Contents

1. Project Description

What if photographers had robotic assistants capturing happy moments in events such as weddings, birthdays or graduation ceremonies? 'Robographers' is the preliminary effort aimed at developing such autonomous assistants that not only click photos, but also recognize and capture the human expressions accurately with equal competency. The principle of the project is facial expression recognition and accurate head pose tracking using a swarm of robots. Instead of working individually, a swarm of mobile robots will work collaboratively to accurately estimate the human expressions and click snaps accordingly.

By using multiple cameras, it is possible to improve the estimate of the facial expressions and head pose through redundant noisy measurements and better handling of real-world issues such as occlusions. By adding mobility to the cameras, thus making each camera a dynamically actuated information source, it is possible to further improve the estimate while improving the tracking of the human as they move through an uncontrolled environment.

The project Robographers has been conceptualized by the common vision of a group of students who believe that robots can perform better when they work in collaboration with each other and also with the humans while at the same time ensuring the safety and comfort of the humans in the environment. The goals of the project are as follows

1. The first component involves evaluating multiple cameras and applying computer vision and sensor fusion techniques to fuse the information from multiple static cameras using ROS and IntraFace[1] to develop a more accurate facial pose and expression estimate. IntraFace is a facial expression recognition software developed by the Human Sensing Lab at the Carnegie Mellon University.

2. The second component of the project will involve designing and building the accurate pan-tilt units for the cameras and incorporating multi-camera head tracking into the developed sensor fusion software.

2. Use Case

"Rohit Dashrathi, the luckiest man in the world, is going to have a big wedding today. And he has got a thought to introduce some new elements to this joyful ceremony.

One of his friends introduced him to the latest product from the Gandhi Inc., a group of three robots which can take candid pictures of happy and smiling people collaboratively. This service can be rented from the company and is charged on an hourly basis.

Hosted in a 216 square feet area of the Sida hall and attended by 50 guests, this wedding ceremony is going to last more than three hours. The situation is mostly out of control for professional human photographers. But Rohit is not worried about it, as he has delegated the photography of this very special day in his life to a very special army of robotic photographers, known as the "Robographers".

Throughout the event, Robographers have become a hot topic of discussion. These amusing robots wearing lovely dresses roam around to add a great charm to the wedding. Adjustable pan-tilt cameras on these units can capture photos of people of all heights, leaving no one unattended. The collision avoidance system keeps all the guests safe. Based on the processing speed greater than 20 Hz, Robographers can detected the human, recognize happy faces and take pictures at the very moment when they smile with joy. The most surprising part is that they are able to take candid pictures without drawing too much attention of the subjects and all of the pictures are very natural and of high quality.

Besides recognizing smiling faces, these robots also remember people's faces and thus avoid taking pictures of the same guest too many times. Armed with the head pose tracking

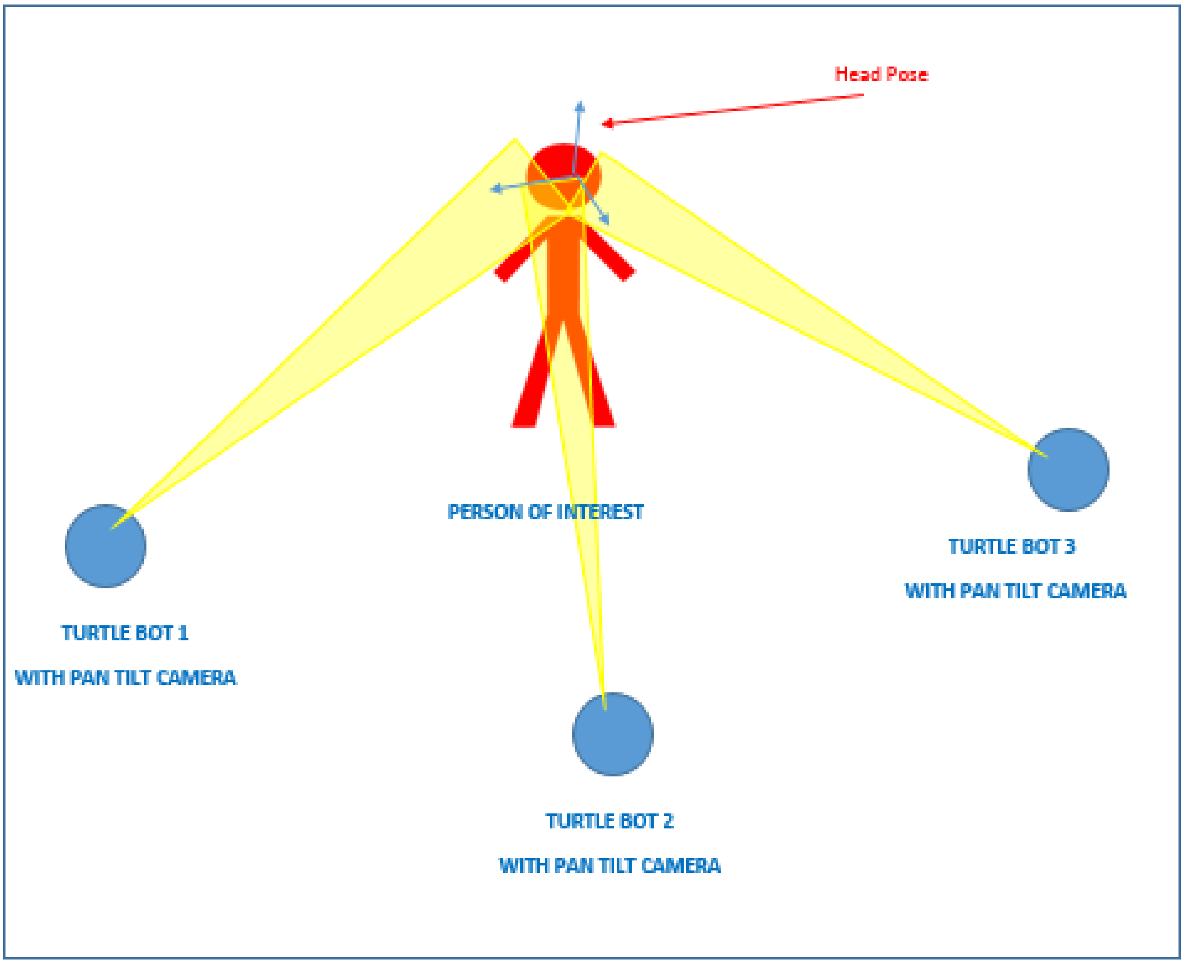


Figure 1 Project 'Robographers': Use Case

technology, the Robographers can tell if the picture is good enough or not by the angle of the faces in the image. The best part is that they can communicate with each other and find all the smiling faces together.

Eventually the robots cover the whole event on both time scale and space scale, which is a high standard where no human photographer can reach. All the guests get natural and smiling pictures in their e-mail boxes sent directly from the Robographers. To Rohit's surprise, the cost of hiring these autonomous photographers is half but the number and quality of pictures is doubled. This is the best wedding ever."

-Article published in Tiffany Times, May 2015

3. System Level Requirements

3.1 Mandatory System Level Requirements

3.1.1 Functional Requirements

Robots in the system shall:

M.F.1: Detect Faces At **0.5s**

M.F.2: Recognise **Smiling** Expression At **0.4s**

M.F.3: Drive Autonomously Between Multiple Locations at **15-20cm/s**

~~M.F.4: Detect Obstacles at **10 Cm** Minimum Height~~

M.F.5: Click Photos In **<1.5 S** Response Time after Expression Detection

M.F.6: Click Photos When Individual/Collaborative Smile Assessment **>50%**

M.F.7: Take Pictures within **3.5-6 Ft** Range

M.F.8: Click At Least **70 %** Smiling Photos (Measure of Overall Performance Requirement)

3.1.2 Non-Functional Requirements

Robots in in the system shall:

M.N.1: Be Supported With Good Lighting Conditions (Fully Illuminated Face All the Time)

M.N.2: Have Wireless Communication mode

M.N.3: Have adjustable elevation

M.N.5: Be easy to operate

M.N.6: Should maintain physical stability (Robots should not topple)

M.N.7: Weigh not more than 11 kg

M.N.8: Should have minimum 3 robots in SWARM

3.2 Desirable System Level Requirements

3.2.1 Functional Requirements

Robots in the system should:

D.F.1: Should identify multiple expressions such as Sad, Disgust, Happy, Scared, Surprise, Neutral, blinks

D.F.2: Should detect multiple faces

D.F.3: Search for human figures quickly

D.F.4: Navigate as quickly as possible

D.F.6: Identify human figures and facial features by same camera

D.F.7: Display photos

D.F.8: Be able to print photos

D.F.9: Work throughout an event (Avg. 4 Hrs.)

D.F.10: Drive autonomously faster between multiple locations At 40 cm/s

D.F.11: Detect Obstacles on the surface

D.F.12: Take Pictures within 20 Ft Range

D.F.13: Click Smiling Photos only (80% Overall Performance measure)

3.2.1 Non-Functional Requirements

Robots in the system should:

D.N.1: Have flash light for good lighting to capture photos

D.N.2: Communicate through Wi-Fi

D.N.3: Have a Graphical User Interface

D.N.4: Not find same person again and again

D.N.5: Have small setup time

D.N.6: Have automatic adjustable elevation

D.N.7: Incorporate 6 robots in swarm

D.N.8: Have minimum 2 Hrs. of battery time

4. Functional Architecture

The functional architecture for Robographers is as shown in figure 2, which can be divided into 3 subsystems: Human detection, planning-navigation & Face-smile detection. It is structured such that the inputs & outputs are on the left and its internal architecture is enclosed in 3 boxes, one for each subsystem on the right hand side.

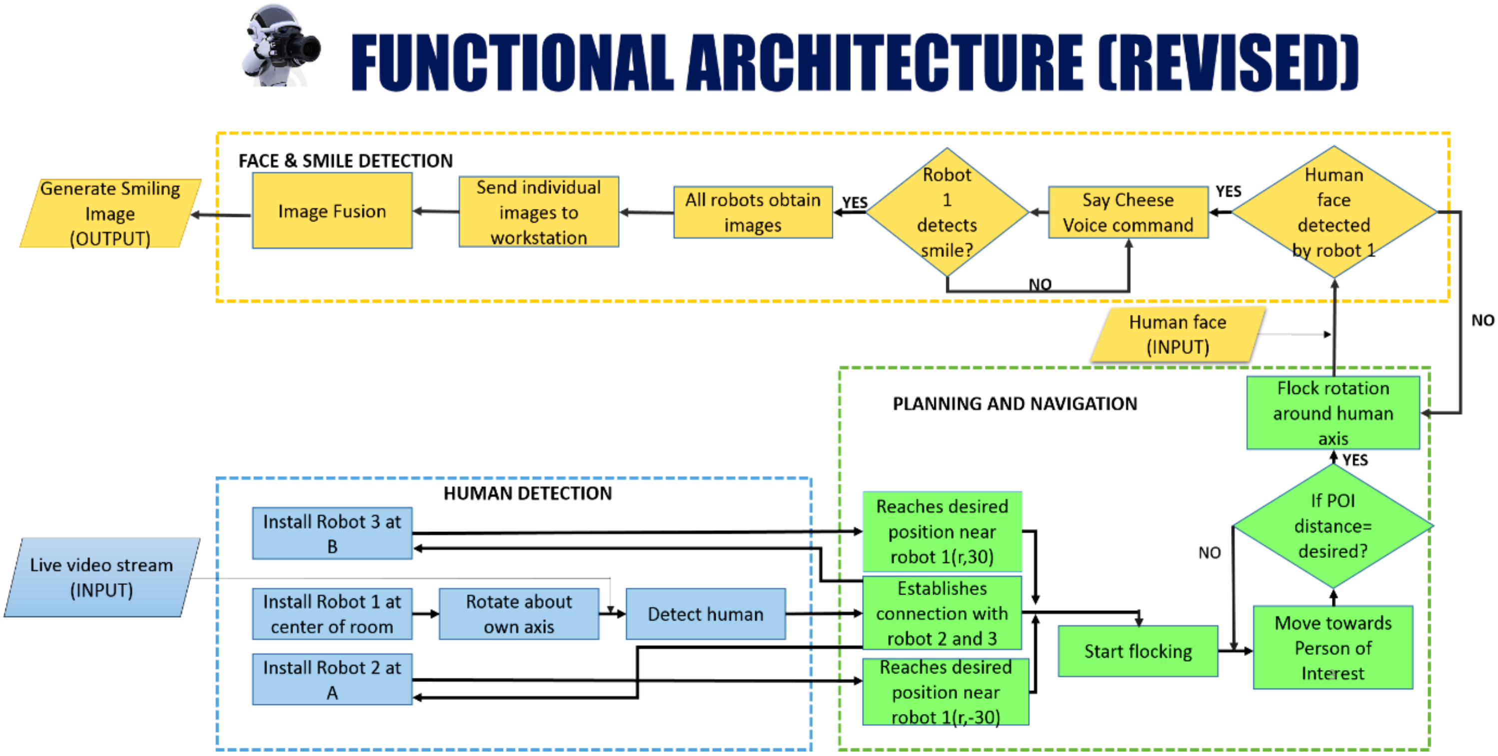


Figure 2 Functional Architecture

Work flow:

1. Turtlebots 1, 2 & 3 will be installed prefixed locations in the specified room. Location for the Turtlebot 1 will be the center of the room while the locations for the robots 2 & 3 will be the any 2 corners of the room.
2. Robot 1 at the center of the room will rotate about its own axis.
3. Web camera on the robot 1 will obtain live video stream (input to the system) of the surroundings in the room while rotating.
4. The camera will look for the persons in the room and will detect the very first person seen through its video feed.
5. Once the desired person is detected, the robot will stop rotating and stay stationary at the position at which it detected the first person. This completes the working of the human detection subsystem. The detected human will be an input for the working of the subsequent planning and navigation subsystem.
6. After human detection, the robot 1 will communicate with the other two robots. Robots 2 & 3 will perform relative localization and will look for the position of the robot 1 using the cameras mounted on them. After detecting the robot 1 position, they will

navigate towards robot 1 at their prefixed relative positions with respect to robot 1.

7. All 3 robots will form a flock and will start moving collaboratively towards the person of interest, detected by the robot 1, performing relative localization.
8. The flock will start moving when the flock reaches a position which is equal to a prefixed desired distance from the person of interest (1 meter). Note that these robots will be oriented in the same manner as decided in step 6. This means that each of the 3 robots will be at 1 meter distance from the human, at angles -30 degrees (robot 2 position, 0 degrees (robot 1 position), 30 degrees (robot 3 position) respectively. Robot 1 will at the 0 degrees position, which means that it will stand at 1 meter from the person of interest, where the person will be facing the camera directly.
9. If the person turns his/her head, the flock will rotate around the human, with the camera on the Turtlebot 1 trying to find the persons face. The system will reorient itself around human at -30 degrees, 0 degrees, 30 degrees respectively as explained before in step 8.
10. After detecting the face, the human will hear a voice request from the system, saying 'say cheese', asking him to smile. Face detection will be performed by the pan tilt camera unit motor movements.
11. When the camera on robot 1 detects the smile and identifies if the smile estimate is greater than 80% (pre-defined threshold), all 3 cameras mounted on the respective robots will click the photo.
12. Images clicked by the robots will be sent back to the workstation. Workstation will identify the common points and perform image fusion.
13. The final output will be the generation of an accurate and clear smiling image.

5. Cyber-Physical Architecture

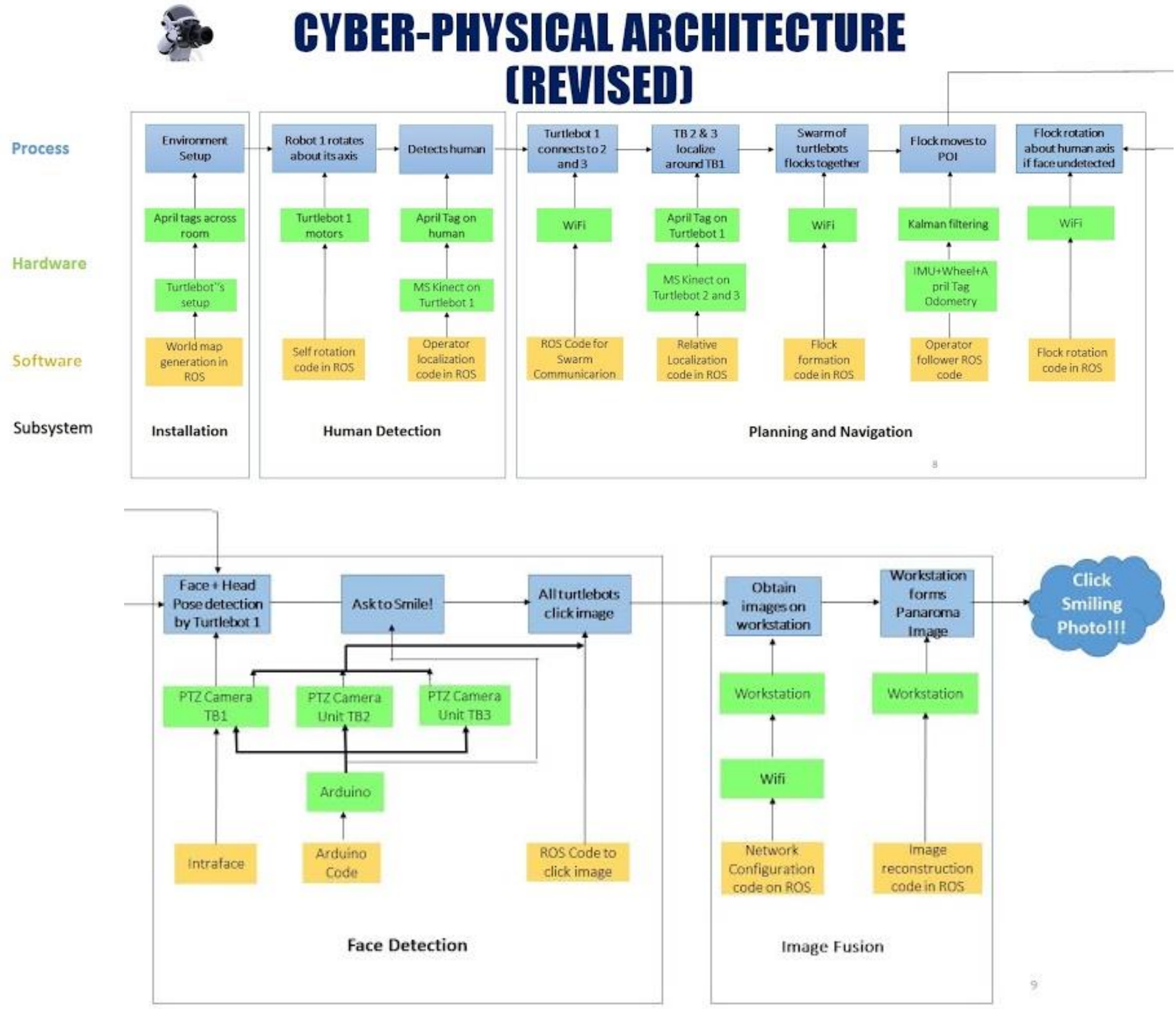


Figure 3 Cyber-Physical Architecture

The cyber physical architecture in complete synchronization with the functional architecture. To make it easier to get the whole idea, the cyber-physical architecture is divided into the same 3 subsystems Human detection, planning-navigation & Face-smile detection. The design of the cyber physical architecture is such that it extends the functional architecture to give a realization of methods/tools used for getting done the functions involved. It depicts the software and hardware components used in each subsystem to complete the tasks expected.

Now we will look at these parts one by one. The installation box is not technically a sub system. It just explains the basic installation procedure before putting the system to work. When the Turtlebots are installed in desired fashion in the specified room, the robot 1 initiates the self -rotation to find the person of interest. All the persons in the rooms are required to carry an April Tag on their shirt for the experiment. The self-rotation of Turtlebot 1 will be carried out using a self -rotation node written in ROS, which will be fed to the Turtlebot motors. The pan tilt unit cameras mounted on the Turtlebot 1 will then try to detect the first person with April Tags on his chest. This will be carried out using the human localization code written in ROS. When the human is detected by the pan tilt camera unit on the Turtlebot 1, the robot will discontinue rotation and will stop at its place. It will then communicate with other 2 Turtlebots over Wi-Fi. This communication will initiate the working of the Turtlebots 2 & 3. They will then localize the linear and angular position of the Turtlebot 1. This will be achieved by the detection of April Tags mounted on the Turtlebot 1, by the pan tilt camera units mounted on the Turtlebots 2 & 3. These 2 Turtlebots, in this way, will estimate the position of Turtlebot 1 and will navigate toward it. They will stop after reaching the desired relative position with respect to the Turtlebot 1. The 3 robots will form a flock in this way, as a result of the flock formation node written in ROS. The flock, as a system, will then start navigating towards the detected person of interest. It will stop navigating once it reaches the desired '1 meter away from human' position. This completes the working of the human detection followed by the working of the planning-navigation subsystem.

When the system reaches desired position with respect to human, the pan tilt unit on the robot 1 will try to detect the face of the human. This will be carried out using the face tracking algorithm written in Arduino IDE format which will be fed to the pan tilt motors through the Arduino Mega microcontroller. If the face is not detected, the system will conclude that the face of the person is turned in some other direction. After this observation, the flock system will perform a rotation around the human, considering human as an axis of rotation. During this rotation, the pan tilt camera unit mounted on the Turtlebot 1 will keep performing its motions to detect the face of the person. The system will stop rotating once it detects the person's face.

After the face detection, the system will try to detect the expressions of the person. The expression detection part will be carried out by the IntraFace software, active only on the computer of the Turtlebot 1. The IntraFace will try to find the expression of the person of interest. When it recognises that the person is smiling, it will try to find the smile estimate as a percentage value. This is an inbuilt facility offered by IntraFace. When this percentage exceed a threshold value of 80 %, the system will issued a command to click photo. This will be done by the combination for the face tracking Arduino code, the face & expression detection IntraFace facility and the photo capture ROS node. The command to capture the photo will be issued to all 3 cameras. It should be noted that the pan and tilt movements of the cameras on Turtlebots 2 & 3 are dependent and hence identical on the pan tilt movements of the camera unit mounted on the Turtlebot 1, which finds and tracks the face. Photos will be clicked by each camera individually when the capture command is issued. The clicked photos will then be transferred wirelessly (using Wi-Fi) to the workstation laptop, not mounted on the system. The ROS code written in workstation will identify the common points of the photos clicked at the 3 desired angles (-30,0,30) and perform image fusion, which is not a separate subsystem, but just an extension of the face & smile detection subsystem as shown in the figure 3. This will yield an accurate and clear smiling photo of the person, which the output of this system.

6. Current System Status

6.1 Current Targeted Requirements

The requirements targeted for fall are all related to single robot – single human system. This will provide an insight of how the system will perform ideally. The requirements are also divided in terms of the three major sub-systems:

6.1.1 Human detection

Functional Requirements

- Detect human
- Design pan tilt unit which tracks the human face

Performance Requirements

- Detect humans with 70 percent success ratio

6.1.2 Planning-Navigation

Functional Requirements

- Drive Autonomously to target

Performance Requirements

- Navigate to the desired 1 meter position and stop
- Drive with speed of 15 cm / second

6.1.3 Face & Smile detection

Functional Requirements

- Detect Faces
- Recognize Smiling Expression
- Click photo

Performance Requirements

- Detect Faces in less than 2 seconds
- Recognize Smiling expression at 0.4 seconds
- Detects face and expression of person within 3.5 to 6 feet height

6.2 Current Subsystem Description

The Robographers system has 3 different subsystems (refer functional architecture) as follows

6.2.1 Detection subsystem

Human detection

The single robot or robot 1 will first have a self-rotation ROS node which will initialize or launch itself at every loop of iteration whenever there is no detected person of interest to be clicked. This node will enable the Turtlebot 1 to rotate in place at angular velocity of 12 degrees/second. The Turtlebot, while rotating, will look for any April Tag (person of interest) in vicinity. As soon as an April Tag is detected, the self-rotation ROS node will kill itself and now there is a person of interest to be clicked.

Facial expression detection

Our main task is to do facial expression recognition with multi-cameras, from 3d direction, instead of the traditional 2d perspective. And it can be achieved by head pose detection, lip angle detection, and eye detection. The software IntraFace and OpenCV will help us in this task. After the swarm of robots reaches in front of the human, next task would be to detect the person's head pose. The robots in system will detect gate and lip of human face for accurate face detection using the camera.

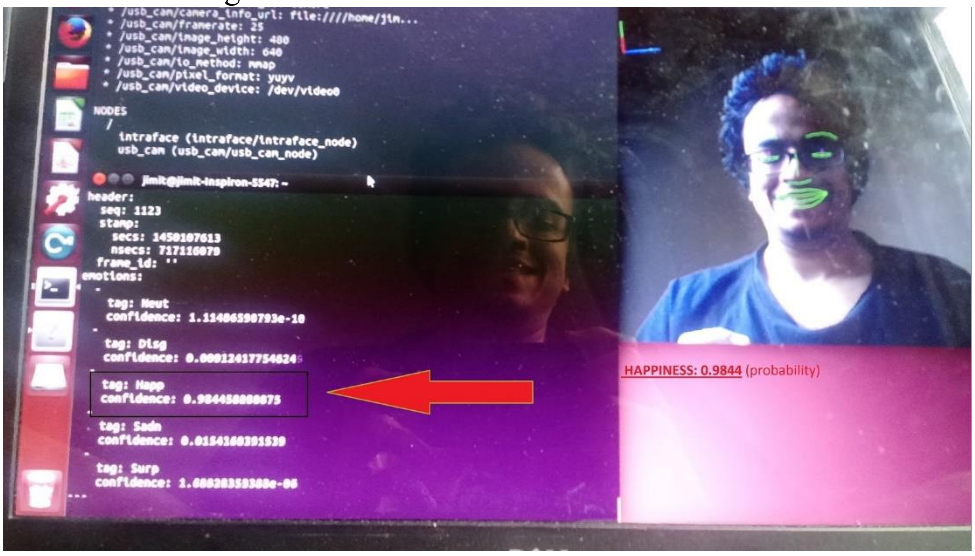


Figure 4 The person of interest is smiling and the "happiness" reading is 0.98

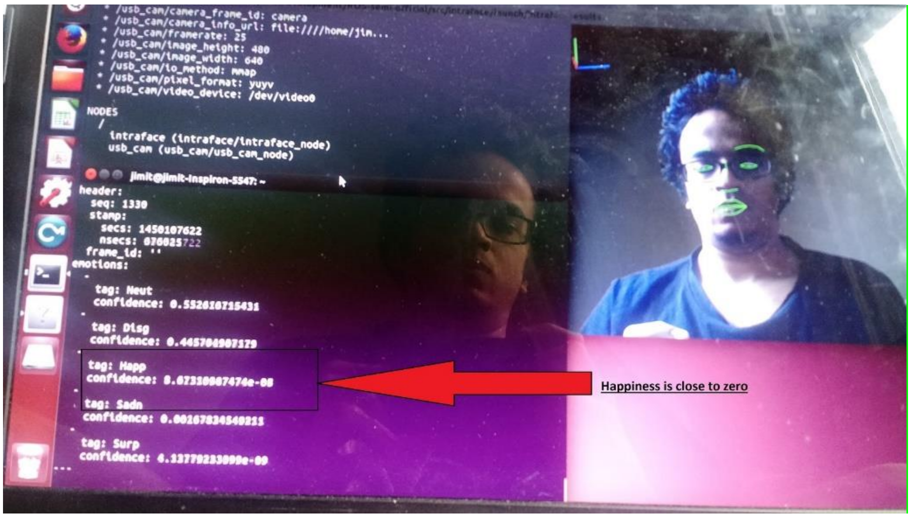


Figure 5 The person of interest is sad and the "happiness" reading is 0.07

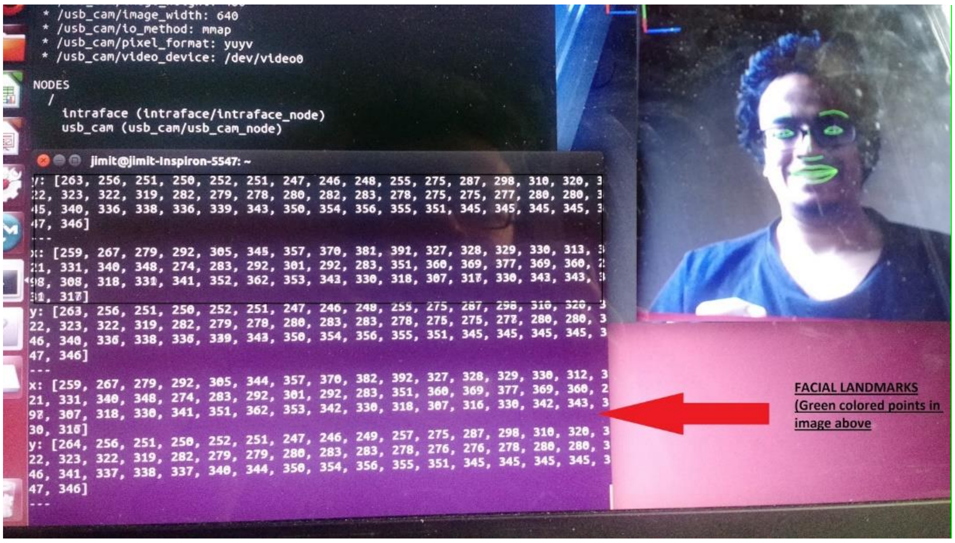


Figure 6 IntraFace gives us the coordinate of 49 interested points of facial landmark as X coordinate vector and Y coordinate vector.

Face tracking control with pan-tilt unit

We track the person of interest by using the coordinate of his/her nose to represent the whole face. The algorithm of face tracking is based on a proportional controller, which covert the width and the length of the image into corresponding angle of pan and tilt servo motors. There's a serious distortion of data range. We constrain the data and convert it to ordinary range using the algorithm similar to Celsius/Fahrenheit conversion. To top the function, we cut the image into three section on the horizontal and vertical direction and form a rectangle of 100*100 pixel, if the nose exceed the region the proportional controller will kick in. Otherwise the motors will remain the same position, thus dramatically reduce the vibration problem of the servo motors.

6.2.2 Planning and Navigation sub-system: Self-Rotation

The project involves the use of an accurate navigation sub-system so that the swarms can easily move through the indoor environment and can place themselves at accurate positions to perform facial recognition collaboratively. There are three basic sub-blocks of navigation-

Calibration

The Turtlebots were each required to calibrate before using. Calibration is a step by step iterative process which should be repeated until the error is zero. The Turtlebot should be placed facing wall on one side and open floor on the other side. There is a ROS launch file that is available on the internet which is used for calibrating Turtlebot. It is a part of the Turtlebot bring-up package. Once "calibration.launch" is launched the Turtlebot starts rotating in place and makes two complete rotations in place. Then it increases its speed by itself and rotates for another round and so on. When all is done, it tries to find the wall again and figure out if the IMU gives out accurate readings and if it does not then it calculates error and prints out correction value. This value has to be manually entered in to the ROS based Turtlebot GUI named "RVIZ." The above process is repeated until we get the correction value close to 1 with a 2-5 percent variation.

Self- Rotation

Turtlebot 1 will self-rotate about its own axis in clockwise sense with angular speed 12 degrees/second. This node will launch itself when there is no person of interest detected. This node subscribes itself to April Tag detector node which will publish April tag information as soon as it detects one. Once the node gets information from April tag subscription, it will call back a function where in it destroys itself and stops the rotation of the Turtlebot. The angular speed is set to a value such that April tag is detected even at a distance of 10-15 feet easily.

Planning and navigation

In the one robot- one human system, we do not require the entire map. Instead we just need the coordinates and orientation of the human relative to the Turtlebot 1. Fortunately the April Tag library provided by Professor Michael Kaess of CMU gives us a quite accurate relative position and orientation of the April tag. The information given by the April tag library is x, y, z and row, pitch, yaw. Additionally it also gives the Euclidean distance. The navigation node we set up uses this information from the April tag and tries to reduce the relative pitch and also reduces relative Euclidean distance to 100 cm. The result is that the Turtlebot 1 is 100 cm away from the human and is facing the human face to face.

The non-inertial motion of the Turtlebot 1 during starting of movement and stopping introduced vibrations on pan-tilt unit which is undesirable. Thus a proportional controller was implemented with desired velocity as 15 cm/second. The integral and differential controller were also implemented but they did not improve the performance much. So we planned to drop it and used only proportional controller. One of the reason being that Turtlebots are already

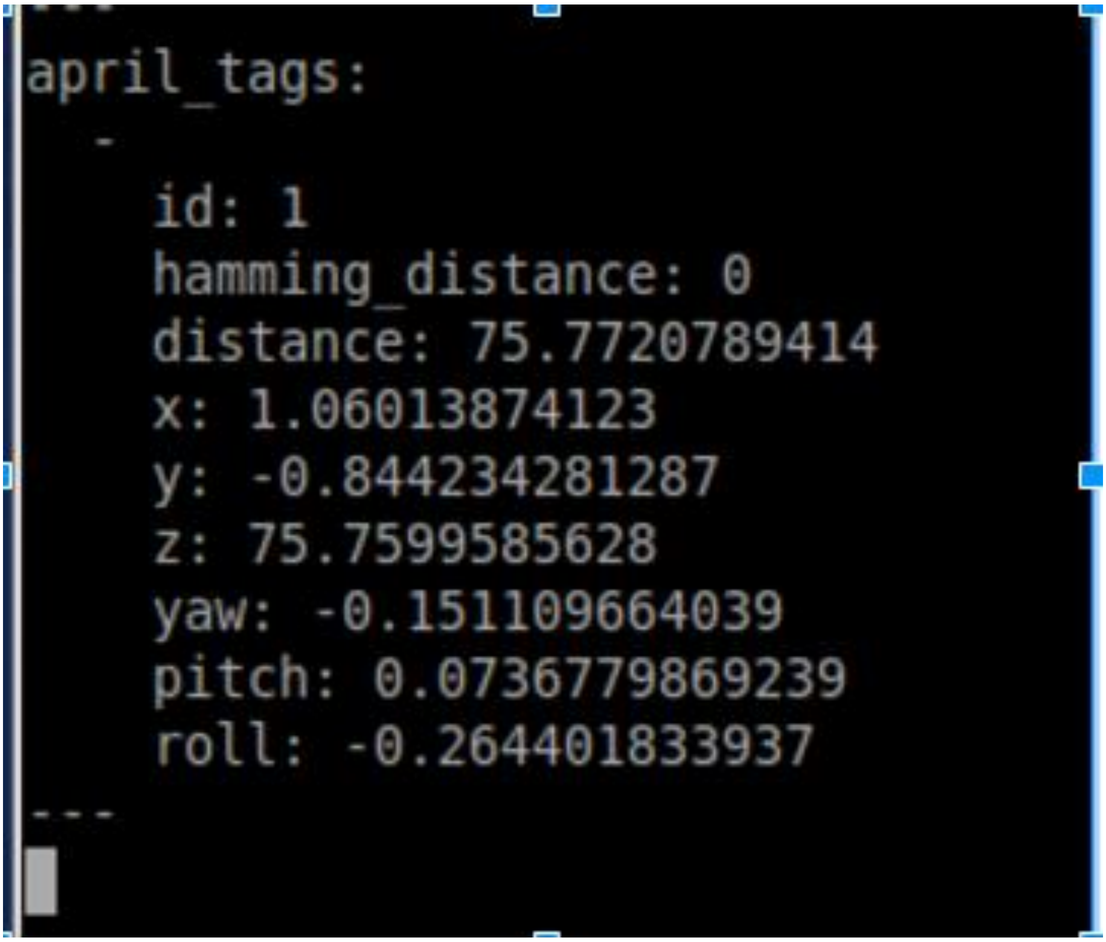


Figure 7 April tag topic output. It prints out x, y, z, yaw, pitch, roll, id and distance values.

6.2.3 Mechanical Subsystem

The mechanical subsystem consists of the pan-tilt elevation axis. The pan-tilt-elevation rod are to be mounted on the Turtlebot. The procedure followed were Design, Finite Element Analysis followed by Fabrication.

Design

While designing the pan-tilt unit + elevation rod system, following things were taken into account.

- Weight of the pan-tilt +elevation rod > payload capacity of Turtlebot. (7lbs)
- Design should be stable such that there should be vibrations on the camera if the Turtlebot is moving
- The elevation rod should be flexible in the sense that we can set the height of the camera according to our will.
- Design should be easy to mount and unmount.

Figure 8 CAD model of Pan-tilt +Elevation rod unit mounted on turtle bot

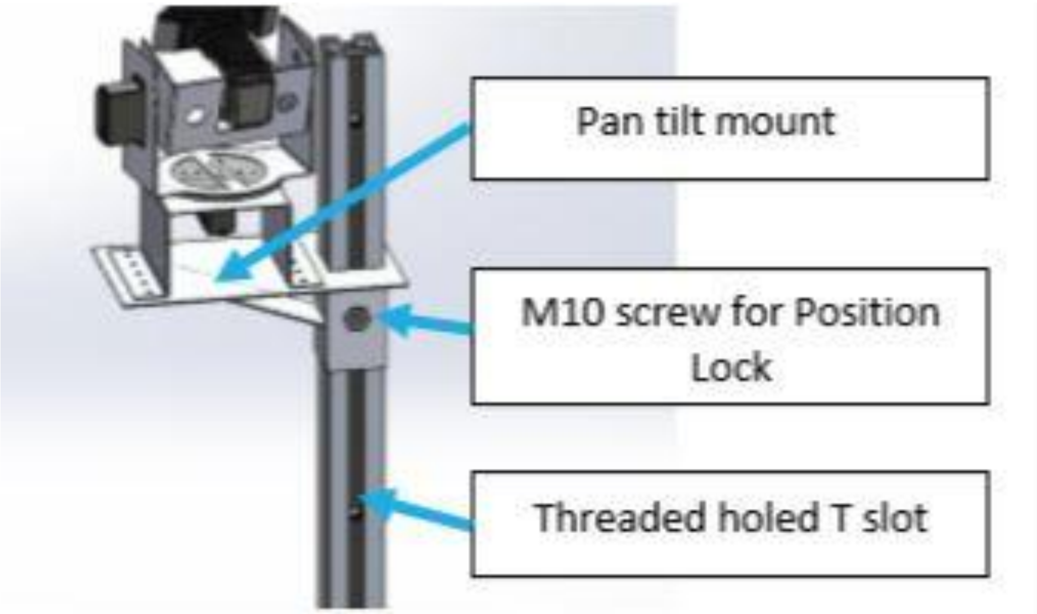


Figure 9 Detailed view pan tilt unit with elevation rod and screw to hold pan tilt at a given height

Finite Element Analysis

After the design part, analysis was done to cross check if the above requirements were met. The first design introduced tremendous vibrations. Using CAD animation tool box and after detailed analysis of the structure, it was concluded that the base of the elevation rod had four critical points known as “failure points”. Once these points were identified, a new design was proposed. On performing similar testing and analysis the vibrations were minimized to a great extent and thus was now within acceptable standards.

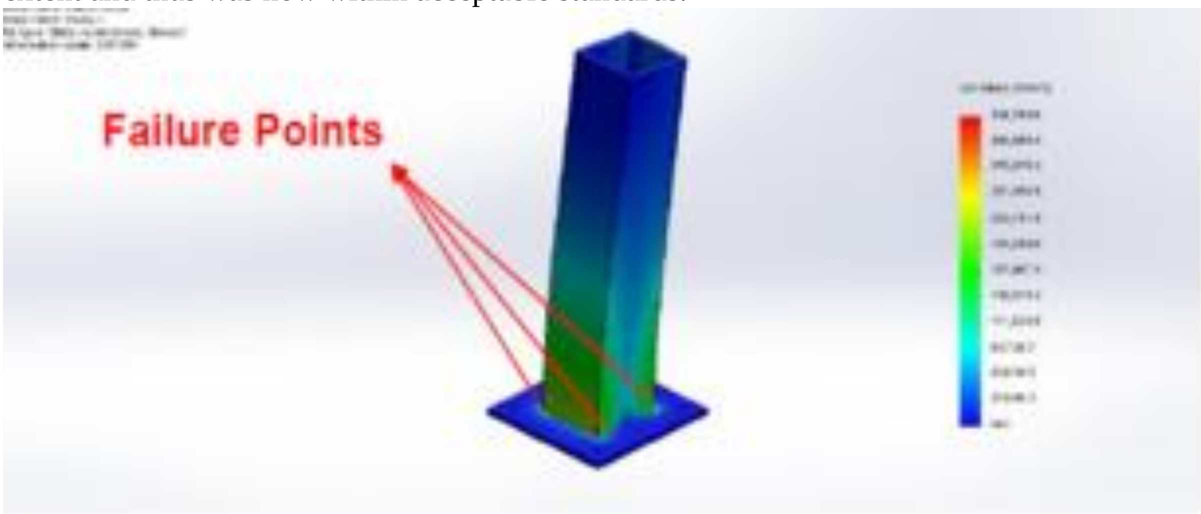


Figure 10 Failure points as identified in the initial design at the base of the elevation rod which connects the rod to the Turtlebot.

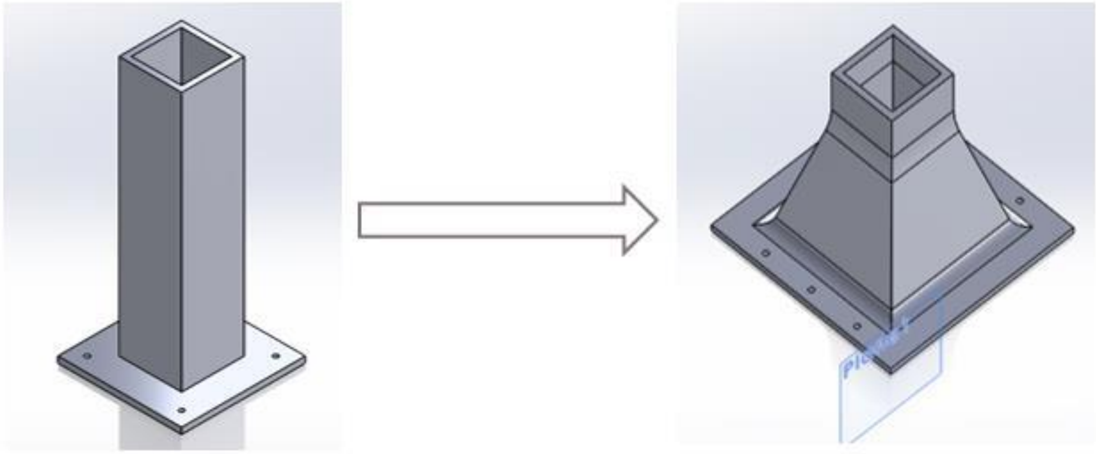


Figure 11 Change in design from initial design which had 4 failure points. The new design is the final workable design.

Fabrication

In order to meet the payload requirements mentioned in the design, it was decided that the proposed designs be fabricated using medium grade aluminum. In order to make a first working prototype as a proof of concept for Fall Validation, the design of pan tilt were fabricated using 3D printed parts while the elevation rod was an aluminum rod available in the MRSD lab inventory. The parts fit exactly and worked exactly as predicted. The fabrication using aluminum will be done in the spring semester.

6.3 Modelling Analysis and testing

The analysis and testing was done keeping the requirements in mind. We rewrite the requirements again. Since the major two subsystems that we were to be verified were

- Human Detection and Navigation
- Face and Expression Detection

6.3.1 Subsystem 1: Human Detection & Navigation

Requirements to be fulfilled

- Detect human in the vicinity
- Approach the human once detected
- Move with a speed of 15 cm/sec
- Stop at 1 meter away from the human

The following questions were framed as which if answered would lead to the fulfilment of the above mentioned requirements. If all of these questions are answered "yes" (in performance) by the subsystem then we could confidently say that the subsystem works.

Testing Criteria for Human detection and navigation subsystem:

- Does the robot rotate in place?
- Does the robot stop rotating in place instantly if April tag is detected for at least 70 percent of the times?
- Does the robot move toward the April tag?
- Does the robot stop at 1 meter away from the Human?
- How much does the final distance deviate from 1m?
- Does the robot stop when it is face to face with the human ?(no relative orientation difference)

6.3.2 Subsystem 2: Face & Smile Expression Detection

Requirements checked

- Detect Face
- Pan tilt unit tracks the face using head pose estimate from IntraFace
- Accurate smile expression detection

Similarly for this subsystem following questions were framed as which if answered would lead to the fulfilment of the above mentioned requirements. If all of these questions are answered "yes" (in performance) by the subsystem then we could confidently say that the sub-system works.

Testing criteria for face detection and expression detection

- Does IntraFace detect face at least 80 percent of the time
- Does the pan tilt unit adjust itself such that face is in center of the frame?
- Do we get expression output every time?
- Time taken to do the above task?

The figures in the next page will verify the questions we framed. For facial expression subsystem the results were verified as shown in Figure 4.

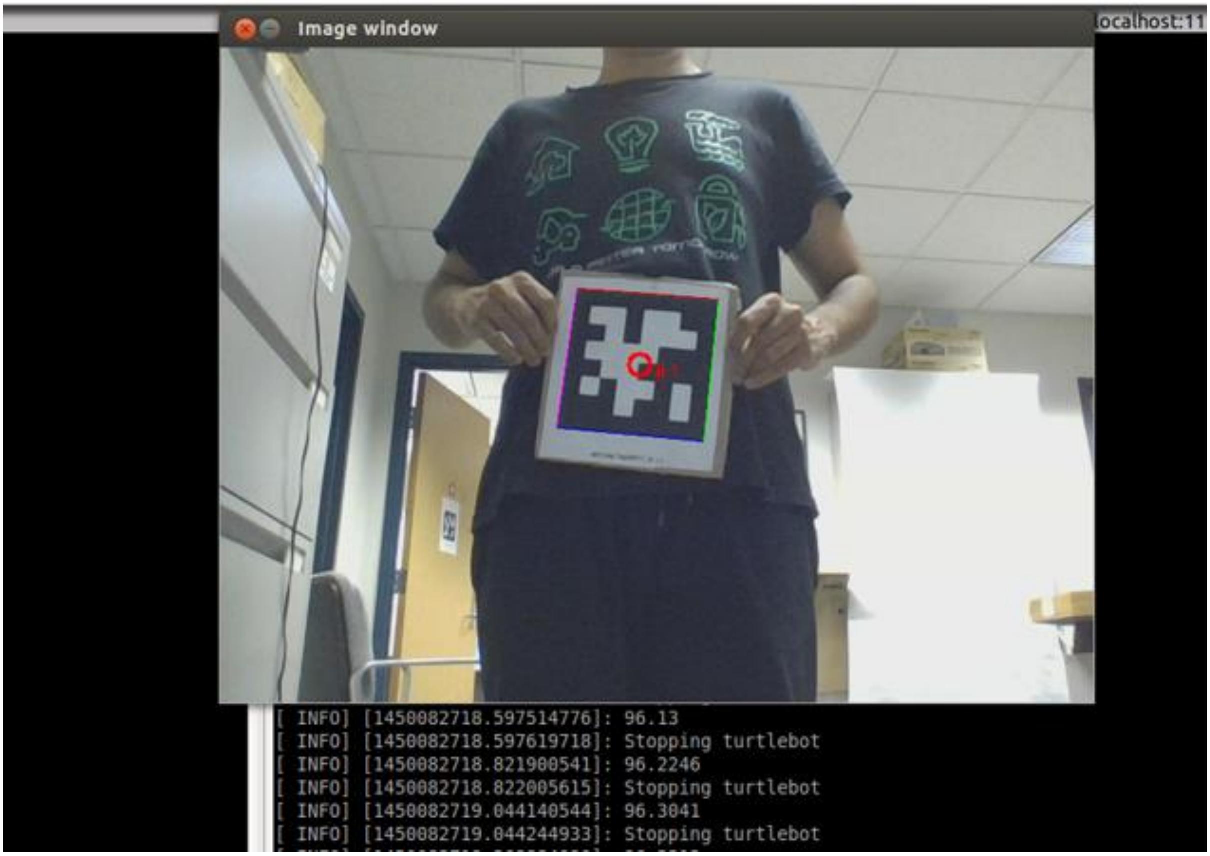


Figure 12 Turtlebot stopped at around 100 cm with an error between 5-10 percent.

Figure 13 Turtlebot successfully detects the human and navigates towards it. Stops at 100 cm away from the human.

6.4 Fall Validation Experiment evaluation

6.4.1 Human Detection and Navigation Subsystem

Based on the questions framed using the requirements of each subsystem in the last section, we prepared a table of criteria which would affect the answers to those questions depending upon the success and failures to meet them.

The table is designed such that if the criteria is met, then we mark it as 1 or we mark it as 0 otherwise. If the criteria is met partially and is within the acceptable value then we mark it as 0.5. The total score of each iteration is taken and divided by the number of criteria. This would give us the success and failure probability. If this probability is more than what we promised in performance requirements then the system is delivered as promised but if not then there is certain improvement that has to be done.

For each of the subsystems we performed 10 iterations with different initial conditions and restarting the system entirely. By different initial conditions, we placed the Turtlebot at random position in the room and human was placed randomly a with the April tag in the room.

System Testing
(Performance Matrix)

Human Detection & Navigation Subsystem: Net Success Percentage=90%								
Run No.	Launched:1 Does Not Launch: 0	Starts Rotation: 1 Does not start: 0	Detects first tag:1 Detects 2nd April Tag: 0.5 Detects nothing: 0	Starts navigation towards tag at 15cm/s	Tracks human in field of view	Stops at : 1m : 1 0.89-0.99 m:0.5 Does not stop:0	Tracks human after stopping	Total (>4.9 success)
1	1	1	1	1	1	0.5	1	6.5
2	1	1	1	1	1	0.5	1	6.5
3	1	1	1	1	1	0	1	6
4	1	1	1	1	1	0.5	1	6.5
5	1	1	1	1	1	0.5	1	6.5
6	1	1	0.5	1	1	0.5	1	6.5
7	1	1	1	1	1	0	1	6
8	1	1	1	1	1	0.5	1	6.5
9	1	1	1	1	1	0	1	6
10	1	1	1	1	1	0.5	1	6.5

Table 1 The performance matrix for the Human Detection and Navigation subsystem.

If we observe, the success of each of the criteria is important. Failure to meet even one criteria could significantly affect the performance of the system. The last column is the total of each iteration and it is out of 7. In order to meet the 70 percent requirement it has to be greater that 4.9 out of 7. The results were calculated as above and when the numbers in the last columns were divided by the total number of criteria, the net success percentage we got was more than 90 percent which is more than what we promised as 70 percent in the requirements.

The Human detection and Navigation subsystem worked better than what we promised so now there is no more scope of improvement required.

6.4.2 Face expression and smile detection subsystem

In this subsystem, as in the Human Detection and Navigation Subsystem we find out the criteria based on the questions we framed in modelling and analysis. Success scenarios of each of these criteria again is important to the fulfilment of the requirement of this subsystem. The scores were allotted in a similar manner as in the Human Detection and Navigation Subsystem. Here too we performed the entire experiment 10 times with different initial conditions. For this subsystem the different initial conditions were changing the person of interest as well his expressions. The person than moved his face randomly in any direction such that the tracking of head would be tested properly.

System Testing
(Performance Matrix)

Face & Smile Detection Subsystem: Net Success Percentage=88%						
Run No.	Launched:1 Does Not Launch: 0	Detects Faces in 2s: 1 Detects Faces in > 2s: 0.5 Does not detect: 0	Tracks faces in pan direction:1 Does not track:0	Tracks faces in tilt direction(3.5-6ft) :1 Does not track:0	Detects Smile in 0.4s: 1 in > 0.4s: 0.5 Does not detect: 0	Total (>3.5 success)
1	1	1	1	1	1	5
2	1	1	0	0	1	3
3	1	1	1	1	1	5
4	1	1	1	1	1	5
5	1	1	0	1	1	4
6	1	1	1	0	1	4
7	1	1	1	1	1	5
8	1	1	1	0	1	4
9	1	1	1	1	1	5
10	1	1	0	1	1	4

Table 2: Performance Matrix of the Face and Smile Detection subsystems

On performing the total as in above subsystem, the promised performance requirements was 80 percent success ratio. Based on this the total should be anything more than 3.5. The overall performance came out to be greater than 80 percent and thus we achieved all our performance requirements.

Thus we delivered what was promised by us in Fall Validation Experiments. In fact we got better results than what was promised so we can confidently proceed with our future development of swarm system and try to complete other requirements as soon as possible. The only requirement that we could not deliver was the photo clicking node after smile detection which will be done first thing in the spring.

6.5 Strong and weak points of current system

6.5.1 Strengths

Integration of subsystems of IntraFace, ROS, Arduino, and Motors: We have implemented the integration of all the subsystems of perception: IntraFace, ROS, Arduino, and motors. The operating system of perception subsystem is Ubuntu 14.04. And we installed 64 bits IntraFace which can be controlled by ROS commands on Ubuntu 14.04. In this way, we can start to do facial expression recognition with IntraFace and simply extract the position of interest points, the percentage of each facial expressions as well as other features with ROS commands. The integration of ROS and IntraFace is very significant because we will have to use ROS to control the Turtlebots' other subsystems. Furthermore, we used Arduino to control motors. The motors can rotate according to the position of the position of person's face. In order to make sure that the smiling person is always in the middle of the camera, we tracked the person's nose's position to make the person's nose always in the centre of the frame. And this can be done with Arduino code. Moreover, we integrated all the four subsystems of perception together. The whole process is as follows: first, we use ROS command to start extracting person's facial expression and interest point's position. Second, IntraFace with ROS transfer the position of the person's nose to Arduino. Third, Arduino calculates the angels that the motor should rotate in both vertical and horizontal directions according to the position of person's nose.

The process is fast without much delay: One of the biggest strength is that the perception subsystem is very fast, and there seems to be no delay for the whole process, which means that the motor can track the person's face and the IntraFace can detect the person's facial expression in a very short time. As our aim is to take photos for smiling person, it's very important to make sure the system to be real-time.

April Tag Library provides accurate readings: The values of the relative pose of April tag given by the April tag library are quite accurate and also provides us with acceptable final results to fulfil requirements

Turtlebot performance is consistent: Turtlebot is a good mobile base as it runs at consistent speed and thus we get desired speed. Also it is easy to calibrate the Turtlebot and it has an extensive library at our expense.

6.5.2 Weaknesses

IntraFace may fail after 7-8 minutes: As there is internal bugs in our IntraFace software, the software may fail after several minutes, which may have bad effect on our project. And to solve this problem, we will contact the IntraFace people to help us.

The perception system is not physical stable: Arduino, Turtlebot and pan-tilt unit would form a perception subsystem, but this system is not physical stable enough that the camera may swag when tracking the person's face.

IntraFace is a Black-box: IntraFace is a black box and we have no control over it to get the desired results. For example we do not have any control which face this software would detect. It does it automatically. All we could do is alter the input such that we get the desired output.

7. Project management

7.1 Work plan & tasks

The work plan for this project is detailed in the work breakdown structure flowchart shown in figure 4. The main categories of the work breakdown structure are based on the work packages indicated in blue colour. For better segregation of work, we have created 2 separate work breakdown structures for each semester. Red coloured boxes indicate the tasks which are not yet started. Yellow boxes indicate the tasks in progress while the green boxes indicate the completed tasks. The overall plan is to develop and validate a single robot photo capturing system in Fall semester, followed by the development and complete integration of the 3 robot SWARM system in the Spring semester. The WBS for spring has not been marked with any colour as evidently no work for the spring semester has been started yet.

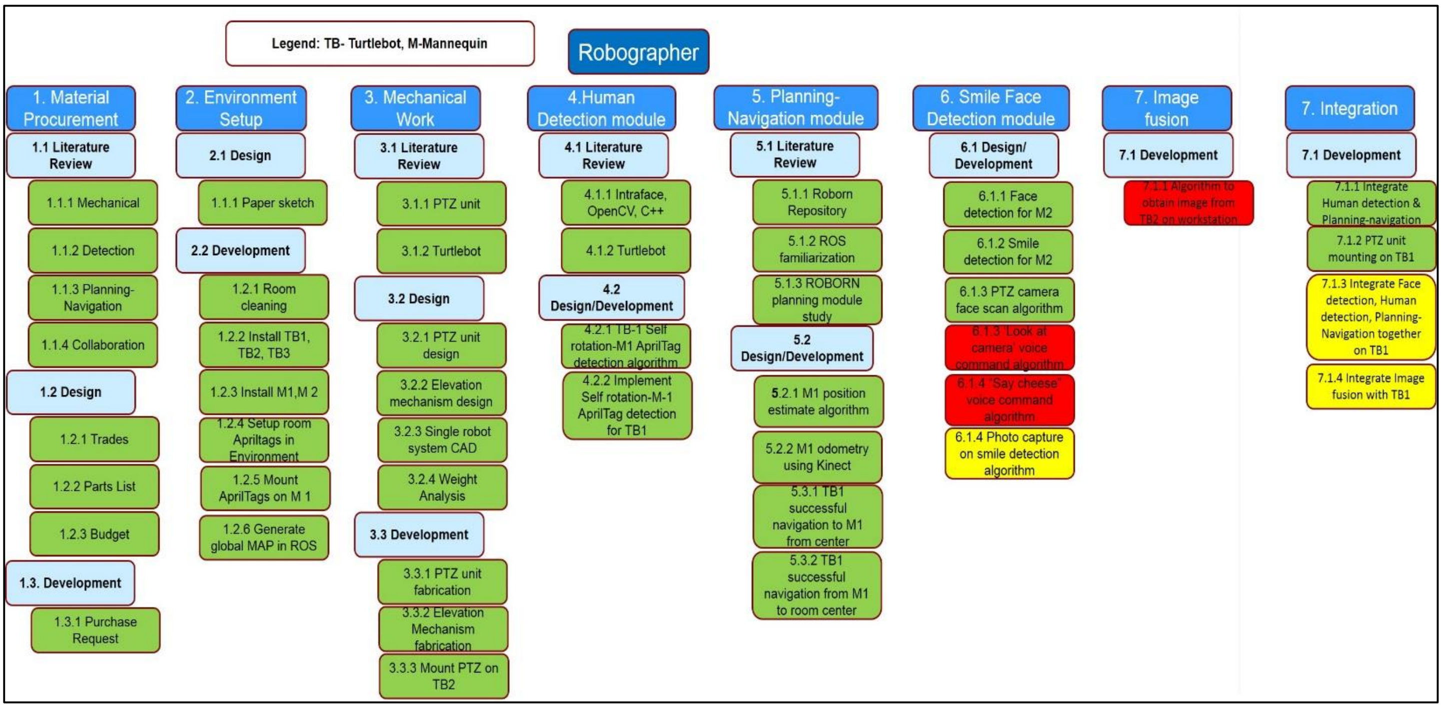


Figure 14 Work Breakdown Structure (Fall 2015)

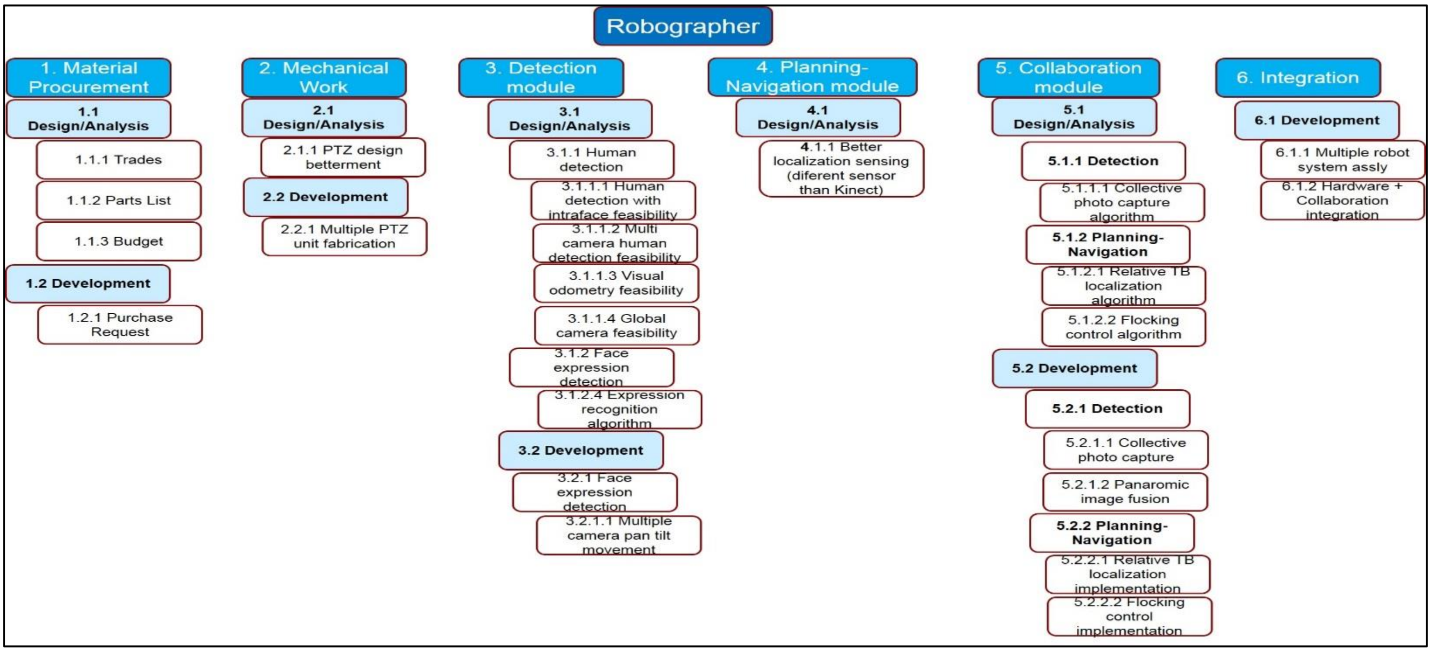


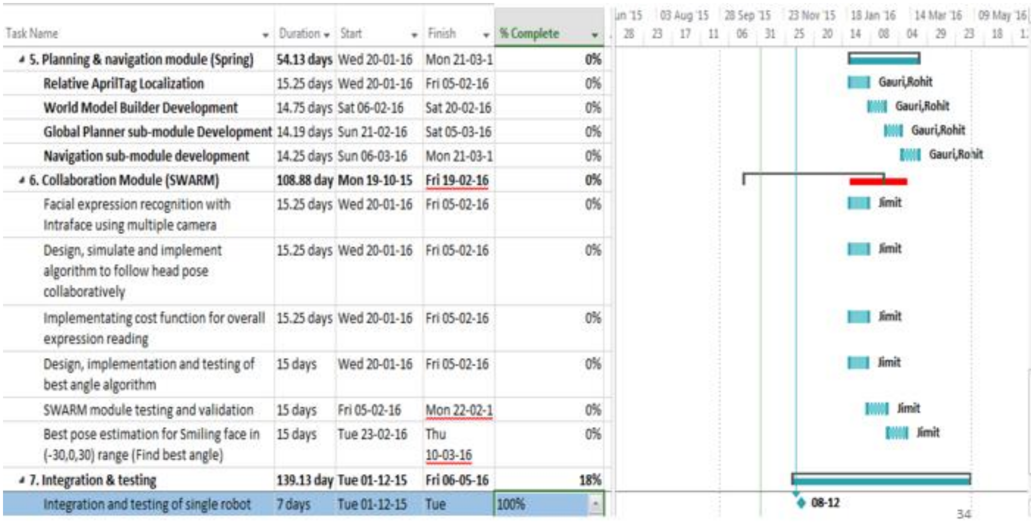
Figure 15: Work Breakdown Structure (Spring 2015)

7.2 Project schedule

We changed our approach on tracking the project schedule. Earlier we tried a few softwares such as Microsoft project, Slack projectmanager.com and some more. However, nothing amongst them turned out to be really user friendly. Also, we observed that the Gantt charts were not of much help during our team meetings. They occupied the half of the screen and resulted in unnecessary confusions being created. Hence, we decided to omit the gantt charts from the tracking sheets. Another change was tracking only the high level tasks and not the minute once. Due to the heavy course load in the MRSD curriculum, it became really difficult to track all the minute details. With these changes, modified our schedule tracking mechanism to using a simple google sheet, shared within the team which is shown as follows:

ID	Name	Responsibility	Duration	Start	Finish	Time of Completion	% Completion
2	**Mechanical Work**						
2.1	Elevation Mechanism CAD design	Rohit	2 days	07-Nov-15	08-Nov-15	11.59pm	100
2.2	Elevation Mechanism fabrication	Rohit	3 days	08-Nov-15	10-Nov-15	11.59pm	100
2.3	PTZ + Elevation assembly on TB2	Rohit	1 day	11-Nov-15	11-Nov-15	11.59pm	100
3	**Human detection**						
3.1	Turtlebot 1 self rotation-Mannequin 1 AprilTag detction algorithm in ROS	Gauri, Jimit	3 days	09-Nov-15	15-Nov-15	11.59pm	100
4	**Planning and Navigation**						
4.1	Mannequin 1 position estimation algorithm	Gauri, Jimit	2 days	17-Nov-15	25-Nov-15	11.59pm	100
4.2	Mannequin 1 odometry using Webcam	Gauri, Jimit	7 days	25-Nov-15	28-Nov-15	11.59pm	100
4.3	Apriltag detection algorithm	Gauri, Jimit		28-Nov-15	30-Nov-15		100
4.4	Turtlebot 1 successful navigation to Mannequin 1 (Accurate 5 Ft distance)	Gauri, Jimit	7 days	30-Nov-15	01-Dec-15	11.59pm	100
5	**Smile face detection**						
5.1	Face detection using webcam & IntraFace/MATLAB for Mannequin 2	Tiffany, Sida	4 days	07-Nov-15	10-Nov-15	11.59pm	100
5.2	Smile Expression detection using IntraFace and Webcam for Mannequin 2	Tiffany, Sida		11-Nov-15	18-Nov-15	11.59pm	100
5.3	PTZ camera face scan algorithm in Arduino and Intraface	Tiffany, Sida		19-Nov-15	25-Nov-15	11.59pm	100
6	**Image fusion**						
6.1	Photo capture on smile detection algorithm in ROS	Tiffany, Sida		25-Nov-15	01-Nov-15	11.59pm	0
6.2	Algorithm to obtain image from TB2 on workstation	Jimit		03-Nov-15	10-Nov-15	11.59pm	0
7	**Integration**						
7.1	Integrate Human detection & Planning-navigation	Gauri, Jimit		23-Nov-15	01-Dec-15	11.59pm	100
7.2	PTZ unit mounting on TB1 & testing	Rohit		12-Nov-15	20-Nov-15	11.59pm	100
7.3	PTZ camera + Face detection algorithm	Sida, Tiffany		26-Nov-15	01-Dec-15	11.59pm	100

Figure 16 Project Schedule for Fall 2015



Task Name	Duration	Start	Finish	% Complete
5. Planning & navigation module (Spring)	**54.13 days**	Wed 20-01-16	Mon 21-03-1	**0%**
Relative AprilTag Localization	15.25 days	Wed 20-01-16	Fri 05-02-16	0%
World Model Builder Development	14.75 days	Sat 06-02-16	Sat 20-02-16	0%
Global Planner sub-module Development	14.19 days	Sun 21-02-16	Sat 05-03-16	0%
Navigation sub-module development	14.25 days	Sun 06-03-16	Mon 21-03-1	0%
6. Collaboration Module (SWARM)	**108.88 day**	**Mon 19-10-15**	**Fri 19-02-16**	**0%**
Facial expression recognition with Intraface using multiple camera	15.25 days	Wed 20-01-16	Fri 05-02-16	0%
Design, simulate and implement algorithm to follow head pose collaboratively	15.25 days	Wed 20-01-16	Fri 05-02-16	0%
Implementating cost function for overall expression reading	15.25 days	Wed 20-01-16	Fri 05-02-16	0%
Design, implementation and testing of best angle algorithm	15 days	Wed 20-01-16	Fri 05-02-16	0%
SWARM module testing and validation	15 days	Fri 05-02-16	Mon 22-02-1	0%
Best pose estimation for Smiling face in (-30,0,30) range (Find best angle)	15 days	Tue 23-02-16	Thu 10-03-16	0%
7. Integration & testing	**139.13 day**	**Tue 01-12-15**	**Fri 06-05-16**	**18%**
Integration and testing of single robot	7 days	Tue 01-12-15	Tue	100%

Figure 17 Project Schedule for Swarm and Planning Subsystems for Spring 2016

7.3 Test Plan

7.3.1 Milestones for Spring Semester Progress Reviews

S.No.	Progress Review	Presenter	Milestones
1.	PR7	Gauri	Full Integration of the single robot system developed in the Fall semester; integrate the human detection, navigation and face tracking and expression detection on a single turtle bot mounted with the pan tilt unit
2.	PR8	Tiffany	Show swarm collaboration using three turtle bots depicting flock formation
3.	PR9	Jimit	Showcase Swarm navigation using the flocking algorithm; improve the accuracy of the odometry for navigation; use april tags for relative localization
4.	PR10	Sida	Accurate swarm arrangement around the person of interest covering a view from -90 deg to +90 deg
5.	PR11	Rohit	Integration of the swarm sub-system with the other sub-systems of the project; testing of the integrated system
6.	PR12	Gauri	Validate all the functional and performance requirements and troubleshooting

Table 3: Milestones for progress reviews

7.3.2 Spring Validation Experiment

Description

A single turtle bot will autonomously detect a human in the environment and send signals to the other turtle bots to form a flock with it. The swarm of robots then moves towards the person and arranges itself around the person. The swarm then tracks the face of the person as well as his facial expressions and clicks a photograph if he/she is smiling.

Location

Advanced Agent Technology Lab (1604), Level 1, Newell Simon Hall.

Set-up

A rectangular empty area of 143 inches* 212 inches. Set up consists of three turtle bots each mounted with a pan tilt camera unit and a netbook with Intel NUC processor. There will be a remote master computer, ie the workstation with one of the operator. At least of the operators will be standing as the persons of interest in the above mentioned area. There will be no obstacles in the prescribed area.

Procedure

1. A lead turtle bot rotates in its place until it detects the April tag mounted on the human.

2. It sends its location and the location of the human to other Turtle bots which will be placed at nearby coordinates(figure 18).

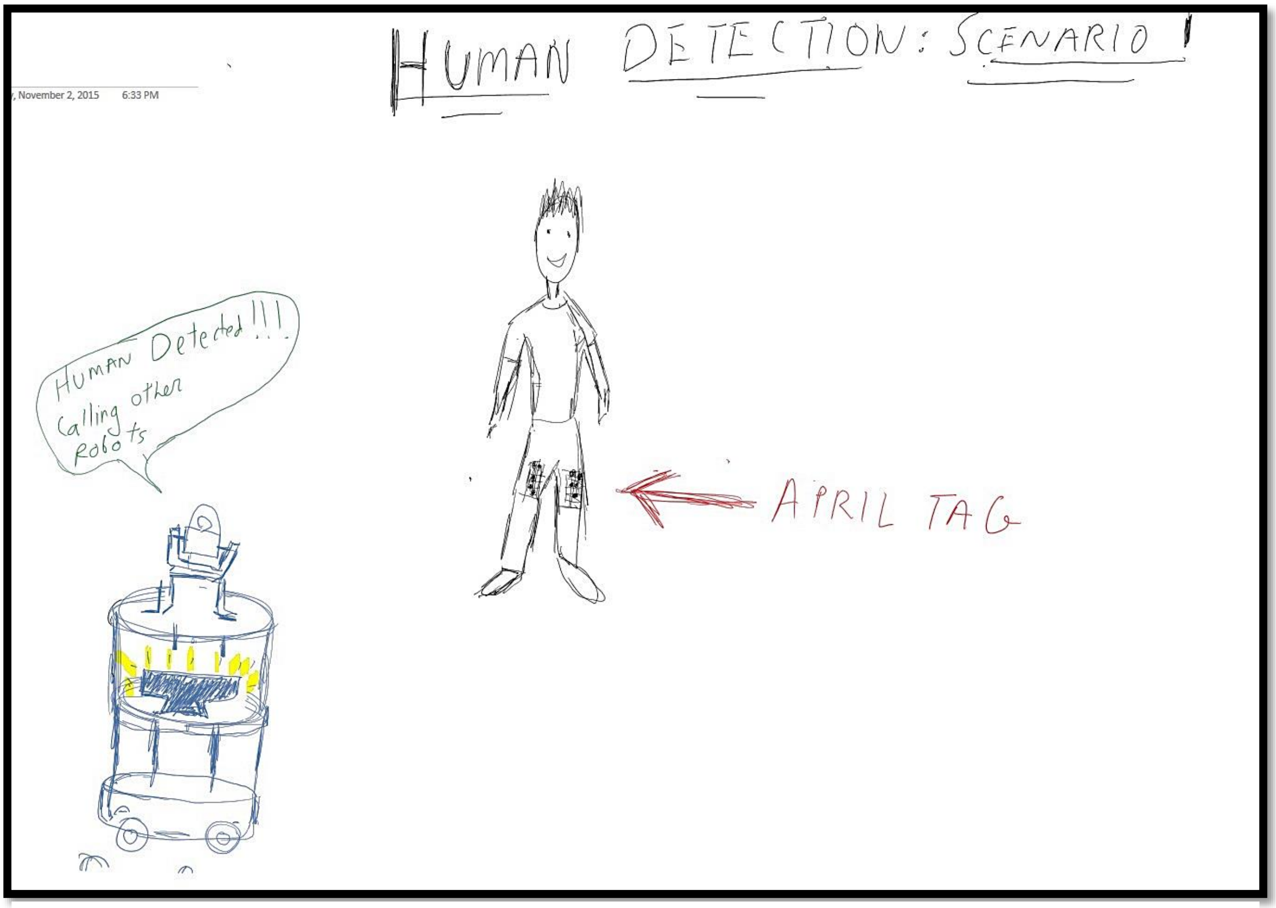


Figure 18 Turtlebot1 detects human using April Tag

3. The three turtlebots form a flock and start moving towards the detected human.

4. The three turtlebots arrange themselves around the human so as to capture a field of view from -90 degrees to +90 degrees. They keep a minimum distance of 1 meter from the human(figure 19).

5. The turtle bots will collectively find the expression reading of the human using the Intraface software.

6. As soon as the human is detected smiling, one of the turtle bot with highest smile reading will click the photograph of the human and send it to the remote workstation.

Success Scenario

1. The lead turtle bot will successfully detect the person with the april tag.

2. It will notify the location of human successfully to other turtle bots.

3. The turtle bots will collectively plan path and place themselves around human at every 45 degree angle from human head pose direction.

4. The robots will simultaneously detect expressions using the Intraface software.

5. They will click photo from the best angle (more facial expression coverage) and send it to workstation.

Functional Requirements to be fulfilled in Spring validation experiment

1. Recognizing expressions collaboratively
2. Plan path autonomously
3. Communicate within themselves
4. Move autonomously from one location to another
5. Track head pose and detect expressions
6. Click photos from best possible angle

Performance Requirements to be fulfilled

Figure 19 Face tracking and expression detection by 3 turtlebots

1. Detect Faces in 2s

2. Recognize Smiling Expression At 0.4s

3. Drive Autonomously Between Multiple Locations at 15 cm/s

4. Navigate to the desired 1 meter position and come to a stop (1 meter away from the human to be photographed)

5. Detects faces of the persons within a height range 3.5 Ft to 6 Ft

7.4 Budget

7.4.1 Parts List

S.No.	Part	Quantity	Price
Mechanical Sub-system			
1.	4 ft long Aluminium extrusions	4	35$ each
2.	Hi-Tech HS 331 servo	8	N/A: Available from MRSD inventory
3.	Arduino Uno	4	N/A: Available from MRSD inventory
Planning and Navigation Sub-system			
1.	Intel NUC5i3RYH Mini PC NUC Kit	4	260$ each
2.	Kobuki Base	4	250$ each
3.	April Tags	10	N/A: Available from Advanced Agents Lab
Face Detection Sub-system			
1.	Logitech C930e HD Webcam	4	105$ each
2.	Intraface software licence	4	N/A: Available from Human Sensing Lab

7.4.2 Budget statistics

Total Allocated Budget = 4000$

Total Budget used in fall/Total budget spent till date = 0$

Total estimated budget to be used in the Spring Semester = 2600$

Total percentage estimated to be used for the project = 65%

7.5 Risk Management

Risk management is very important since this tool helps us to foresee the problems that may arise and how they will affect the project performance and functionality. As it is rightly said "Hope for the best but prepare for the worst", risk management helps us prepare for the worst. The team identified the potential problems that could occur given the current status and marked them as malignant or benign based upon the consequences and likelihood of that problem if it occurred. The following are the potential risks identified and also the corresponding strategies or steps proposed in order to reduce and mitigate the effect of that risk. The risks have also been pointed out in figure 7 in the form of risk management chart.

1. **Risk #1: Noisy and blurry video stream due to robot moving.**

Risk Type: Severe: Likelihood is more since robot will be always moving and capturing video. Since expression detection involves clear video stream to capture the micro-facial interest points it is pertinent that vibration caused to the camera due to robot movement be minimized so as to get high efficiency of expression detection software.

Risk mitigation strategy: The source of vibration was due to the less strong base of the elevation axis. The design of the base was modified to make it stronger and firmer. In this way the vibrations were reduced significantly. The risk in now benign as the system can now perform without much difficulty and thus meeting the performance requirements.

2. **Risk #2: IntraFace Software crash**

Risk Type: Severe: Since the prime functionality of the system is expression detection, the working and robustness of the IntraFace software is a must. The likelihood of this risk as well as consequence lead to undesirable effects and ultimate failure of the system. The likelihood was as high as 6 times crash in 10 test runs. The mitigation of this risk was a high priority.

Risk mitigation strategy: A more stable ROS version of IntraFace was obtained from the Human Sensing lab at CMU. The new version seems to crash less (1 time in 10 test runs) and is promising. If the software crashes at rare occurrences, one can program the node to launch itself again. Although it is not a good thing to initialize the same node again and again but if it is done at low frequencies then it is acceptable.

3. **Risk #3: Battery drain in Turtlebots.**

Risk Type: Moderately severe: The battery life of Turtlebots is less (30-40 minutes from full charge), repeated testing cannot be done. This is a wastage of the most precious resource of our team which is time. We have faced events that lead to frustrations.

Risk mitigation strategy: The reason that Turtlebots have low battery life is because the Turtlebots we are working on currently are 5 years old. This has reduced the battery life considerably with repeated timely charging and discharging. The lab sponsor has decided and ordered a new set of batteries as well as spares such that in the event of low battery, one can replace a battery and put that discharged battery on charging. Our efficiency has doubled since the arrival of the new batteries.

4. **Risk #4: Single Robot failure**

Risk Type: Moderately severe: If a single robot fails then the performance of the system goes down. Although the likelihood of this risk is relatively low, the consequences could be high. Even if other robots work perfectly fine, failure of a single robot could lead to undesirable effects in terms of performance.

Risk mitigation Strategy: The swarm subsystem should be planned and executed such that in the event of a single Turtlebot breakdown, the performance of the system as a whole should not be affected to a great extent. We could have one or two spare Turtlebots in the inventory (which we do). Also another approach we plan to implement is, in case of such a failure and no availability of the spares, the system should do perform the task intended using whatever the number of Turtlebots available with a slightly reduced performance.

5. **Risk #5: Extra Payload**

Risk Type: Moderate: The Turtlebots have a limited payload capacity of 7-9 lbs and beyond that they will not achieve the speed and movement we desire. Thus it is imperative that the weight of the entire pan- tilt plus the camera with elevation axis should not exceed the above mentioned limit.

Risk mitigation strategy: There are many light weight HD camera available in the market which are within our budget as well. We have included weight of camera as an important parameter during camera trade study analysis. The pan tilt design also is designed such that we use minimum material but at the same time provide support and stability to the mounted camera. The material to be used will be aluminium.

6. **Risk #6: IntraFace as a black-box.**

Risk Type: Moderate: The IntraFace software is the state of the art software developed by Human sensing group at CMU and is the fruit of 5 years of research. The understanding of how their program works is beyond our knowledge horizon. This can be a problem since we cannot control it in any way such that we have the output we desire for example, detecting the expression of the person we specify.

Risk mitigation strategy: While understanding the entire code will be counter-productive, we plan to coordinate with human sensing lab members and understand only the part that we want such that we are able to tweak certain parameters. We can also make changes in the system such that we can maximize the control over the input for example, perform image processing techniques over video stream such that we input only the face of the person that we want expression of and ignore the rest as background.

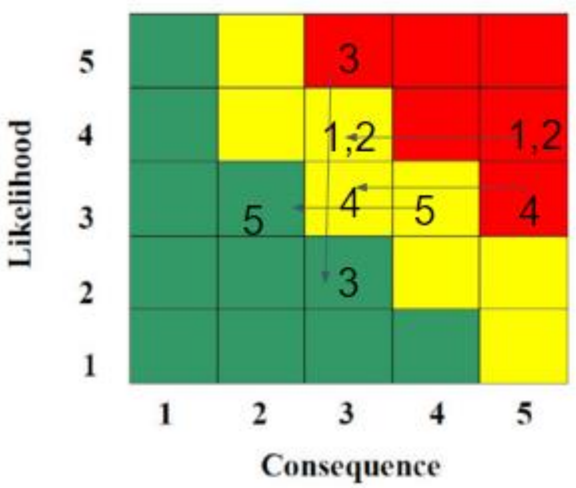


Figure 20 Risk management table: The above figure depicts the numbers of the risks mentioned in the description on previous page. The arrow suggests that the risks were initially where the arrow starts and where the risks now stand after mitigating them, represented by the arrow heads.

8. Conclusions

8.1 Lesson learned

1. For those very important things, always prepare a plan B

Prepare a plan C for plan B, because everything may crash down. If nothing crashes, something will automatically crash to fill the void.

2. Communication

Always communicate with your teammates. Don't expect others will know what you are thinking. And don't start doing your job without letting other people know what you are doing. Sometimes not doing something is as important as doing something.

3. Try to see the future. Find out the risks

Then you can prepare early to face the risks.

4. Do not expect magic from day 1

Magic wouldn't happen and the future will not be changed if we don't change our behavior. We learnt to be practical and start things as early as possible. After starting, we realized all kind of odds could stack up against us. Some of the problems didn't reveal themselves unless we started, like the unexpected crash down of Intraface and noise problems we faced in moving data.

5. Keep everyone on the same page

The miscommunication happened at the very first time of the team meeting. Five people had five different ideas. And since we tend to not speak out and disagree with teammates, it took a while until we finally reach the agreement. But it was worth it, after we were on the same page, we could reach the goal at the full speed without hesitation.

6. Clarity is very important

The clarity is the key point of whether the communication is successful or not.

7. Do not reinvent the wheel

This lesson was learnt when we were trying to build our own human detection and face tracking function, and then we realized that Intraface already did that for us. If we would have found out that earlier, we could have saved some time.

8. Take small steps to achieve required goals

Some of the work seems impossible to be done. Cutting down tasks to small sub tasks always helps.

8.2 Spring activities

8.2.1 Shor-term goals

1. Integration of all Subsystems developed in Fall

For now, the sub-systems for a single robot system have been developed independently. The pan-tilt unit is connected to the Intraface. We will connect this subsystem to the turtlebots navigation and human detection subsystem in the spring.

2. Define and implement algorithm for accurate smiling

We are considering using the percentage of smile and combining the position of turtlebots to implement the algorithm of whether the person of interest is smiling or not. These readings will be sent to master computer which will select the photo with maximum percentage of the person's smile according to the positions.

3. Implement the clicking function

We will add the clicking function and capture photo with the current webcam, which is the webcam we use for facial expression detection.

8.2.2 Long-term goals

1. Swarm coordination and communication

We will be implementing the complete swarm subsystem in the Spring semester.

2. Navigation and path planning by multiple robots

The three turtlebots will communicate with each other and flock autonomously towards the person of interest using the April tags.

3. Image processing of video such that input to Intraface is desired human face

Right now there is no control over the person being detected by the Intraface. It detects randomly from a given frame. We will work on to develop an algorithm that selects the closest person from the given frame and feeds that to the Intraface.

4. Multi-face recognition

We will work on an algorithm to recognize more than one faces at the same time.

9. References

Project Report Format: https://sites.google.com/site/mrsdproject201415teamc/

1 http://www.humansensing.cs.cmu.edu/intraface/

2 http://wiki.ros.org/rocon

3 http://shop.soloshot.com/

4 https://en.wikipedia.org/wiki/Lidar

5 http://www.roborealm.com/forum/index.php?thread_id=2681

6 http://www.amazon.com/Panasonic-HC-V110-Weight-Digital-Camcorder/dp/B00AW54YWS