Firefighter Indoor Navigation Development Plan

Client:	Mostafa Daneshgar Rahbar
GTA:	Samira Taghavi
Team Members:	Thomas Anter Nawar Mikha Jordan Shimel

Table of Contents

Revision History

Date	Version	Authors	Comments
9/13/19	1.0	Thomas Anter Nawar Mikha Jordan Shimel	First Draft
9/16/19	2.0	Thomas Anter Nawar Mikha Jordan Shimel	Added revision history, formatting, gantt chart
9/18/19	3.0	Thomas Anter Nawar Mikha Jordan Shimel	Finalizing formatting after presentation feedback
12/5	4.0	Thomas Anter Nawar Mikha Jordan Shimel	Added list of Figures, minor text revision throughout the document

List of Figures

Overview

Firefighters have one of the most dangerous jobs in the world. There are few professions where people are literally risking their lives like firefighters do. When there are people trapped in burning buildings, these professionals are their only hope. As technology excels exponentially, firefighting lags behind in adopting new tech to make the profession safer and more efficient. This project focuses on tailoring new technology to support the needs of this job.

One of the many difficulties of fire rescue missions is dealing with low visibility in an unfamiliar and dangerous setting. It is difficult for a firefighter to make use of hand operated devices while on duty. Having data collected automatically by a camera mounted on the firefighters person, which then transmits the data to a command center, can help with this challenge. On-site commanders need to know which areas of the building are burning and where the team members are in relation to the fire. Having a real-time system located within the fire truck reading data from the firefighter's cameras will help assist the fire department in saving lives.

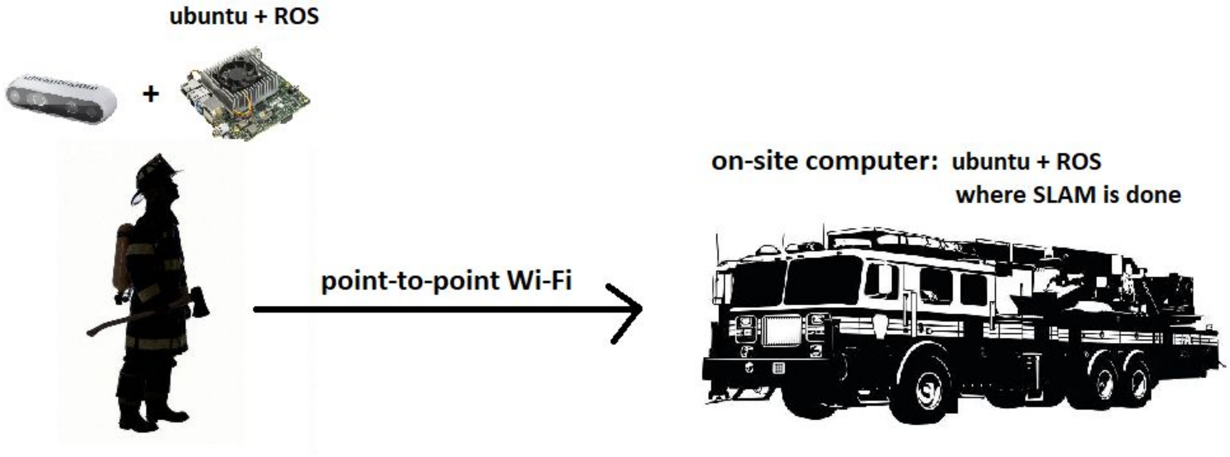

Figure 1

Purpose, Scope, & Objectives

Currently, GPS is used to track the current position of firefighters in an unfamiliar building. This system lacks the ability to digitally tag areas, if the origin of the fire is found, if an area is deemed dangerous, or if people are thought to be trapped in a room. GPS is not sufficient to communicate with other team members if a firefighter needs to go to the location of a teammate. GPS only gives information about the position of the teammates in relation to one another, it will not map the path the firefighter took to get to that point. Most importantly, the reliability of the GPS signal indoors in many buildings is low, and without proper signal, leaving the location of the team unknown.

The purpose of this project is to develop a more efficient way of locating firefighters and guiding them throughout a building. Using an on-person stereo camera, this project will create a digital map of the building and track the path from the entry point into the building, to the firefighters current position. This will give the operator a better visual to not only have a more reliable source of information regarding the location of the team, but will know the path to get to that point. This will allow firefighters to navigate much better on the job.

The long term scope (outside of this semester) of this project is to allow for an operator to use augmented reality to use the mapping from the cameras to navigate the firefighters. Our team will focus on a subset of this scope, and will allow the next team to easily add more functionalities.

The short term scope, and the objective for this semester are as follows:

- The digital mapping and path tracking of the camera in relation to the layout of the building
- point-to-point wif-fi network between a depth finding camera and UP board to a main console laptop
- A GUI with the main console which displays the camera feed and digital mapping

These are the main deliverables of this project. Additionally, we'll need software to allow the camera to read the mapping data, track the path of the camera, allow point to point wi-fi connection between the UP board and main console, and a GUI on the main console to monitor all of the above.

Team Organization

Each member of the team will allocate a considerable amount of time to ensure each iteration is met by the expected deadline. While everyone will be part of the process for each part of the development, every individual will have different areas of focus for specialization and dependability purposes. Listed is a delegation of these roles and a summary of their responsibilities.

Nawar Mikha

- Team lead
    - Responsible for managing team member workloads, project disputes, and mediating interpersonal conflicts
- Presentation lead
    - Responsible for main design and direction of team presentations
- UI lead
    - Responsible for user facing design and interface elements

Thomas Anter

- Documentation lead
    - Responsible for ensuring that documentation is accurate and up-to-date
- Backend lead
    - Responsible for collecting and transmitting data from remote unit(s)
- Frontend lead
    - Responsible for collecting and interpreting data sent by camera

Jordan Shimel

- Testing lead
    - Responsible for designing and implementing code tests
- Integration Lead
    - Responsible for ensuring front-end, back-end, and UI systems function properly together

Conflict Resolution Policies

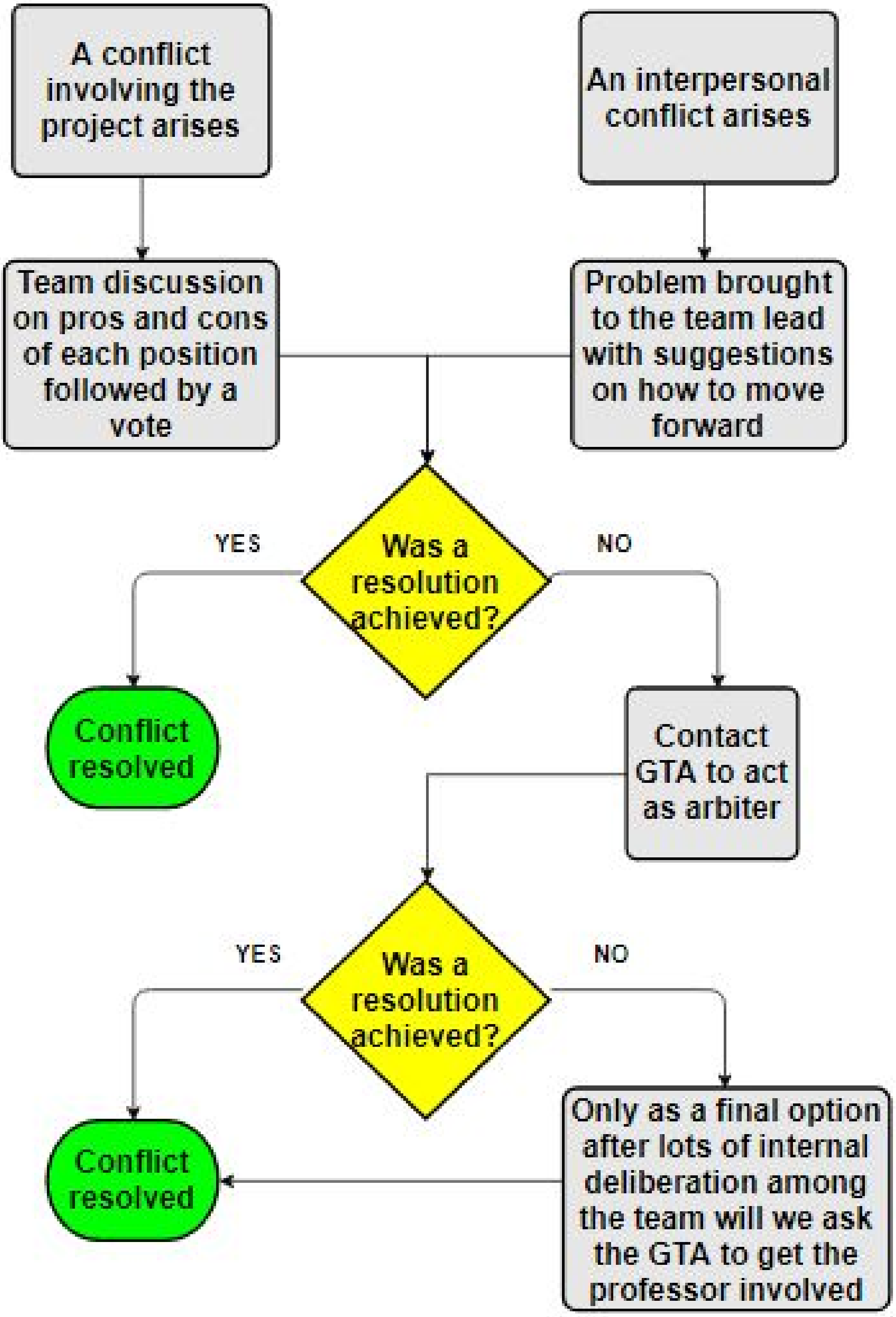
A conflict involving the project arises
An interpersonal conflict arises
Team discussion on pros and cons of each position followed by a vote
Problem brought to the team lead with suggestions on how to move forward
YES
Was a resolution achieved?
NO
Conflict resolved
Contact GTA to act as arbiter
YES
Was a resolution achieved?
NO
Conflict resolved
Only as a final option after lots of internal deliberation among the team will we ask the GTA to get the professor involved

Figure 2

Project Schedule

We have discussed with our GTA and client that we will be meeting with them each individually at a regularly scheduled time every week. Every Thursday we will be meeting with Samira Taghavi, our GTA, at 4pm, and with Mostafa Rahbar, our client, at 5pm. We have determined that we will be meeting every Wednesday in person before class, we also plan on having provisional meetings on Saturday or Sunday when we deem it necessary, whether it's in person or with Skype. Our main form of communication is a group chat using WhatsApp, this is the first place we go when we need to contact the others immediately.

Our schedule is derived from our adoption of an agile development plan. An early working prototype is very important; there will be a working version 1.0 capturing videos with the camera and UP board, then delivering new features with each iteration. This was deemed to be the most efficient method, especially with the possibility of changing requirements. If the scope of the project changes, it will not be detrimental to the development.

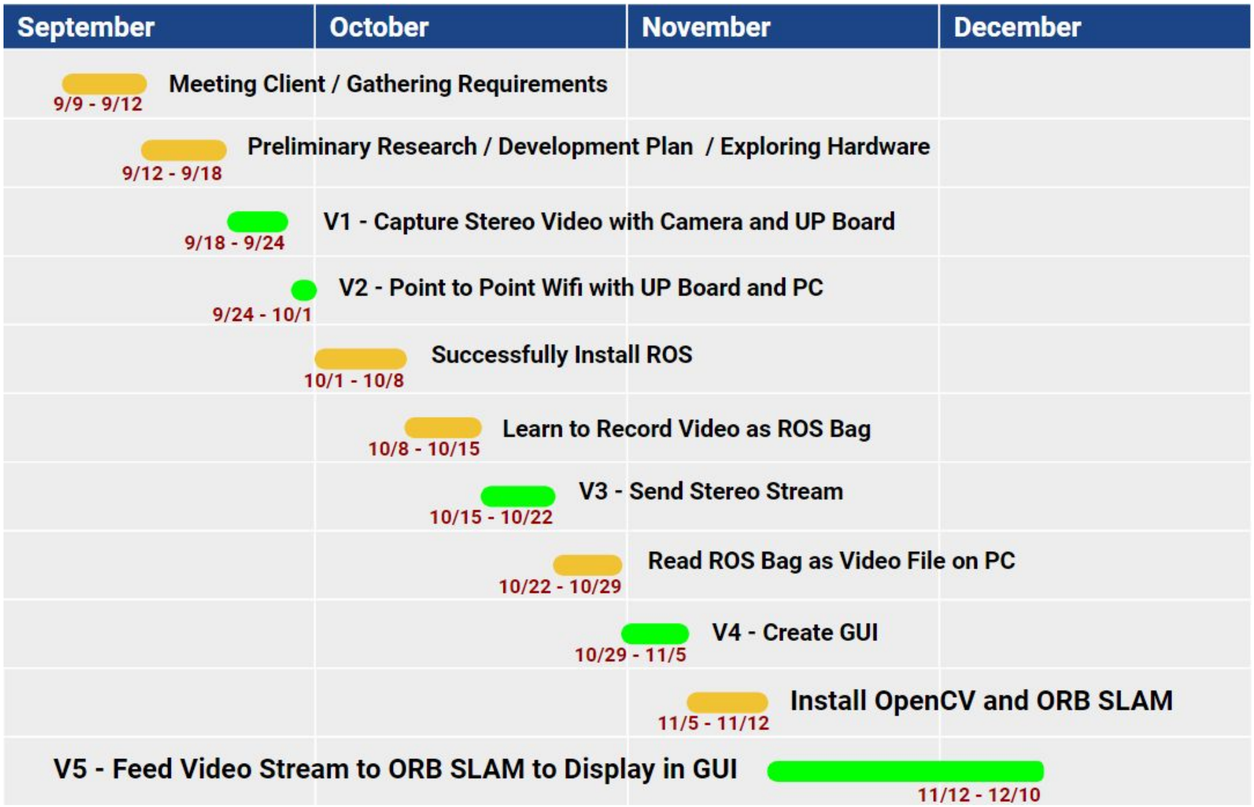

Figure 3

Dates listed are the due dates for the task(s)

9-17-2019

1. Exploring Realsense Depth Camera D435i and its features
2. Studying about the concept of depth camera and particularly D435i and SLAM
3. Installing the camera and communicating with a computer
4. Exploring UP board and its features

9-24-2019 Version 1.0

Write code to capture the stereo video from the Realsense camera with UP board

10-1-2019 Version 2.0

Setup a Point to Point Wi-Fi Network for communicating between UP board and PC to establish a network

10-8-2019

Install ROS, Robotic Operating System, on Ubuntu on PC and UP board

10-15-2019

Learn how to record video as ROS bag on UP board

10-22-2019 Version 3.0

Send the stereo video stream from the UP board to the PC

10-29-2019

Read the ROS Bag as Video file on PC

11-5-2019 Version 4.0

Create GUI on PC to show the stereo video stream for 2 firefighters

11-12-2019

Install OpenCV and ORB-SLAM2 and other Packages on PC to do Simultaneous Localization and Mapping

12-10-2019 Version 5.0 (Final Product)

Feed the video stream to the ORB-SLAM2 package and show the result in GUI

Configuration Management Plan

Each lead will be responsible for breaking down the weekly tasks involved with his assigned aspect of the project. The task for all areas including documentation, integration, presentation, front-end interface, etc… will then be divided amongst the team members based on the estimated time required to finish each task and each individual's skills by the team lead. While one group member may specialize in one area, all of us will be involved in every part of the project.

We plan on following the principles and practices of Agile development for this project. We have all of our weekly goals already set for the next three months. We see our development plan as a living document that may change as we make progress on the application. Our weekly meetings on Wednesdays are when we will have our scrums where we break down our tasks for the following weeks goals or sprints. Wednesdays will typically be when we will be presenting what we have done in the past week; this is the perfect time to delegate the following week's tasks as it will give us the most amount of time to get them done. We will also be delegating tasks on Saturdays or Sundays when we have meetings on those days, as that is when we'll be doing our weekly reporting.

For our source code, we'll be using Github for version control and Doxygen for code documentation. Branches in the repository will be based on the developer; each developer will have his own branches for different tasks he will be working on. Branch names will follow the following format for naming "developerName_assignedTaskDescription".

Technologies

CMake -

Software build tool

DBoW2 -

Library for transforming images into bag-of-words (Requirement of ORB-SLAM2)

Eigen3 -

Linear algebra library (Requirement of g2o)

FFMPEG -

Video capture and encoding (Requirement of OpenCV)

g++ -

C++11 compatible compiler

G2o -

Graph optimization library (Requirement of ORB-SLAM2)

Git -

Github for version control

Glew -

Extension for OpenGL (Requirement of Pangolin)

GLFW-

Extension for OpenGL

GTK+ -

Toolkit for creating GUI (Requirement of OpenCV)

Intel RealSense SDK 2.0 -

API for interfacing with D435 camera

OpenCV 4.1.1 -

Computer vision library for image processing

OpenGL -

Graphics API (Requirement of Pangolin)

ORB-SLAM2 -

Real-Time SLAM for camera data

Pangolin -

Visualization and UI (Requirement of ORB-SLAM2)

PCL-

Point cloud processing library

Qt5 -

GUI development tools

ROS Kinetic -

Middleware for robotics, used for live processing of camera data

Ubuntu 16.04 -

Operating system for UP Board and base station

XDo -

C++ library for window manipulation