CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

DIPLOMA THESIS

Tomáš Rouček

Sensor fusion for object localisation in adverse conditions for mobile robots

Department of Computer Science

Thesis supervisor: **doc. Ing. Tomáš Krajník, PhD**

May, 2020

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Rouček Tomáš** Personal ID number: **434976**
Faculty / Institute: **Faculty of Electrical Engineering**
Department / Institute: **Department of Control Engineering**
Study program: **Cybernetics and Robotics**
Branch of study: **Cybernetics and Robotics**

II. Master's thesis details

Master's thesis title in English:

Sensor fusion for object localisation in adverse conditions for mobile robots

Master's thesis title in Czech:

Senzorická fúze pro lokalizaci objektů v obtížných podmínkách mobilním robotem

Guidelines:

Design and implement a multisensory system to detect and localise objects transmitting radio, chemical, and audiovisual signals in adverse conditions of disaster scenarios.
1) Learn about the Robot Operation System (ROS) and Husky A200 platform.
1) Research sensor technologies and their performance in adverse conditions.
2) Research sensor fusion methods for object localisation.
3) Choose a suitable sensor fusion method, implement it and perform its preliminary experimental validation.
4) Integrate the methods into a robotic system for autonomous multi-robot exploration of disaster sites.
5) Select a set key of performance indicators (KPI) of the intergrated system.
6) Perform validation of the method performance in a realistic scenario (DARPA SubT or MBZIRC) and gather data for later evaluation.
6) Desing and implement an automated evaluation pipeline using the gathered datasets and KPIs.
7) Evaluate the impact of the individual sensors on the overall performance of the deetction and localisation system.

Bibliography / sources:

[1] Pfeuffer et al.: Optimal Sensor Data Fusion Architecture for Object Detection in Adverse Weather Conditions. ArXiv 2018.
[2] Filip Majer et al.: Learning to see through haze: Radar-based Human Detection for Adverse Weather Conditions. ECMR 2019.
[3] George Broughton et. al.: RFID-Based Object Localisation with a Mobile Robot to Assist the Elderly with Mild Cognitive Impairments. Intelligent Environments 2016.
[4] Gustafsson et. al.: Statistical sensor fusion. 2010.
[5] Darpa SubT rules - https://www.subtchallenge.com/resources/SubT_Challenge_Tunnel_Rules.pdf
[6] MBZIRC 2020 rules -
https://www.mbzirc.com/download-document/document/MBZIRC2020-Challenge-Descripti
on-V4-14JAN2019.pd
[7] Robot Operatng System – ROS, http://wiki.ros.org/

Name and workplace of master's thesis supervisor:

doc. Ing. Tomáš Krajník, Ph.D., Artificial Intelligence Center, FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **14.01.2020** Deadline for master's thesis submission: **22.05.2020**

Assignment valid until:
by the end of summer semester 2020/2021

doc. Ing. Tomáš Krajník, Ph.D.	prof. Ing. Michael Šebek, DrSc.	prof. Mgr. Petr Páta, Ph.D.
Supervisor's signature	Head of department's signature	Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt

Student's signature

Declaration

I hereby declare that I have completed this thesis independently and that I have used only the sources (literature, software, etc.) listed in the enclosed bibliography.

In Prague on............................ ....

Acknowledgement

I would like to thank my thesis supervisor doc. Ing. Tomas Krajnik, PhD. for his insight to writing a thesis, all people form CTU-CRAS-NORLAB team for putting up with me when the system did not work as it should and Ana Panaiot for not breaking up with me since I had no time to spare to dedicate to her.

Abstrakt

Cílem projektu je navrhnout komponentu robotického systému pro vyhledávání zraněných při průmyslových a přírodních katastrofách. Hlavním problémem při vyhledávání objektů a jejich lokalizaci je neschopnost běžných senzorů fungovat spolehlivě při nepříznivých okolních podmínkách. V této práci je prezentován systém, který je navržen pro vyhledávání mobilních telefonů a úniku plynu v podmínkách, kde standardní senzory selhávají. Senzorická data z robotické platformy jsou využita ve spojení s pozicí robota za účelem odhadu pozice hledaného objektu v několika různých prostředích, za použití dvou různých metod. Mapa detekovaných objektů a multilaterační systémy jsou popsány ve větších detailech a následně oba implementovány. Systém je následně implementován na systému pozemních robotů pro účely mezinárodní soutěže DARPA SubT Challange. Byl následně využit v nepříznivém prostředí opuštěné atomové elektrárny kde se osvědčil a zajistil teamu CTU-CRAS-NORLAB třetí místo.

Abstract

The goal of the project is to design a component of a robotic system for search of survivors of industrial or natural disasters. The main problem during a search for objects and their localisation is an inability of common sensors to work reliably in adverse condition. This work presents a system that is designed for search of mobile phones and gas leaks in conditions where standard sensors fail to deliver proper measurements. Sensoric data from the robotic platform are used in conjunction with the position of the robot to estimate the location of the sought after objects in several different environments with the use of two different methods. Map of detected objects and multilateration systems are described in greater detail and later implemented. The pipeline is then implemented into a robotic system for the purpose of international DARPA SubT Challange competition. It was used in adverse conditions of an abandoned nuclear power plant where it proved itself and got the third place to CTU-CRAS-NORLAB team.

Contents

List of Figures

1 Intro

When it comes to disasters, humankind must be able to respond quickly and safely, to prevent loss of life. Since human resources are limited, we cannot just mobilise several million people every time there is a flood, earthquake or a mine collapse. Sometimes the accessibility is the issue, or sometimes it is just not worth the effort, but mostly the limitation is that the rescuer can quickly become rescued if he is not appropriately trained. This situation can occur, even if the rescue team took all precautions, but just did not anticipate another cave-in or leaky gas. At that moment is where usually robots come in to play since they are disposable unlike trained professionals.

Usage of robots in search and rescue (S&R) situations is much more extensive than one might guess on the first look. Even though the ideas of a fully autonomous S&R team that needs no supervision are already somewhat researched, the insight into this topic is not large enough, to be deployed in real situations [2]. Most robotic systems, if they are even used, focus on the sole objective. These objectives include, but are certainly not limited to, mapping, reconnaissance, delivery, faster search, better perception, moving rubble, taming fire and much more. Only very few projects ever focused on teaming of robots with even with assisted control [3, 4].

One of the more prominent usages is mapping and reconnaissance, as it saves time for the human rescue team that does the manual job and warns them about potential hazards. To have a map of the area, building or forest helps a lot, even when it is just a basic information, for example where wall are located. More data in the map helps even more, especially if this data corresponds to any object of interest such as a survivor under debris or an IED device placed by terrorists. Those maps can then be fed to mission command or just a rescue team in the field which can make and improve their decisions based on those data.

The current status is that somewhat autonomous search is possible only in specific scenarios which mostly can be tackled by flying robots[4]. Usage of autonomous drones with GPS allows searching a much larger chunk of the ground than a human on the foot could. This system still usually includes an input from the operator which has to assign parts of the maps to be searched, leaving still some load on him. The biggest problem comes down to that usual drone surveillance is not enough since you need to go closer to the ground or eventually inside a building where the luxury of prebuilt infrastructure such as GPS is nonexistent. Another big unsolved problem is the wireless limitations which do not allow for underground or long-range missions.

In those scenarios, robots need to be driven by an operator or have a different way of localisation. If the operator drives the robot, you are not saving the human resources since one can not effectively drive more then one robot and you are running into problems that happen the moment communications between robot become limited. Once the communication is lost, jammed, or you just driven out of the range of the transmitters robot cannot be controlled anymore. In those situations, the most basic solution is for the robot to retrace its steps back, but this approach hinders the speed at which the robotic system helps. Not to mention that having a different system such a robot and an operator, as opposed to just one more rescuer, is much more logistic heavy for any operation. New advancements in technology allowed to tackle this problem in several ways, such as new sensor types or advancement in algorithms and computer technology which allows for a lot of data to be computed onboard each robot to conserve the amount of data needed to be sent back to base.

Recent improvements in sensory equipment and research in robotic prompted Defense Advanced Research Projects Agency (DARPA) to organise an event called Subterranean Challenge (SubT). This challenge is supposed to push and incorporate all those new sensors and algorithms together into a working system in the form of a contest between the best robotic teams worldwide. The goal is to explore underground environments with difficult and hostile conditions, using said advancement in technology.

This event showed that many algorithms that proved to work flawlessly in laboratory environments have to be refined, since they are not mature, to exhibit reliability or robustness in a general environment. One of the problems in this SubT contest is to find and precisely localise given objects. Mostly this is done using visual detection in the image, but some objects have more signatures than only the appearance. For example, a survivor might give off a heat signature, or radio-enabled devices give out measurable radio signals.

The goal of this work is to devise, implement and evaluate a system that helps to find some of the objects such as mobile phones using a robotic platform. This work firstly explains how autonomous systems work and considers sensory possibilities for such a system and describes the pros and cons of different approaches. Thesis then describes the system which was implemented and finally describes results that were achieved during the SubT Urban circuit, which took place in February 2020 in an abandoned nuclear power plant.

2 Robotic sensors, fusion, localisation and mapping

Robotic systems for any autonomy need to have at least a basic idea of how their surrounding is arranged. Without this information, they can only execute preset commands without any feedback or autonomy. This means a robot needs to have a map of his surroundings, as well as sensory data. With its system robot is able to correlate known maps with current measurements from sensors to asses the position and other states about the outer world.

2.1 Sensors in adverse conditions

Sensoric data are nice and crisp when they are tested in laboratory conditions, but when it comes to real-life use, they often fall short. This hinders any efforts of most algorithms, so there comes a need for identifying what the problems are.

2.1.1 Light sensors

One of the most used passive sensor in robotic these days are cameras. Those devices use an array of light-sensitive electronic sensors called pixels which captures the light intensity and converts it into data. They briefly expose light onto this sensor, opening its shutter, which allows for more precise control of the exposure for the sensor. Cheaper and widely used rolling shutter cameras where the shutter does not expose the sensor all at once but rather does sweep across it have a problem when anything they are targeted at moves quickly. This results in a distorted image since some parts of the sensor have the incoming data at different times. The most significant problem is that this also happens when the robot moves since it changes position of the camera making the image move quickly. A second significant problem of conventional cameras is if the light does not come through the medium it is looking at received data are unusable. This comes down to obscuring cameras vision by fog to just not having a light around results in no data gathered. All of those problems affect even stereo cameras which can calculate object distance by looking at it with different angles.

Next big problem with opaque surroundings is that active light sensors such as Depth cameras (RGBD) Fig:1 or Lidar technology needs to send its own light to the surrounding. RGBD cameras function on the same principle as stereo cameras, but they have their own projector of a pattern which they can detect and calculate the depth with higher precision. On the other hand, LIDARs Fig:2 use laser beams that they sent out and measure when they return when they reflect from the measured object. None of those methods is viable when the surrounding is impenetrable by light or scatters it so much one cant identify single beam as in fog or morning haze [5],[6]. Another problem of those sensors is that they assume that the light they shine on the measured object is reflected at least to some degree back at them. Needed amount of reflected light varies from 1% to 20% but always assumes

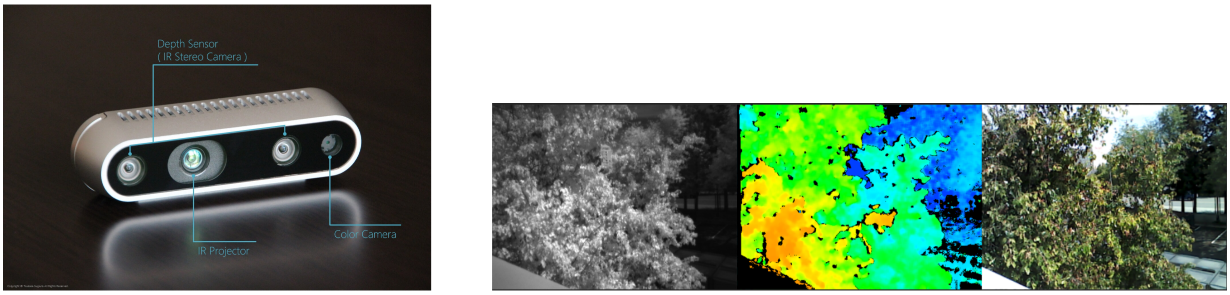


Figure 1: RGBD camera RealSense D435 and its data

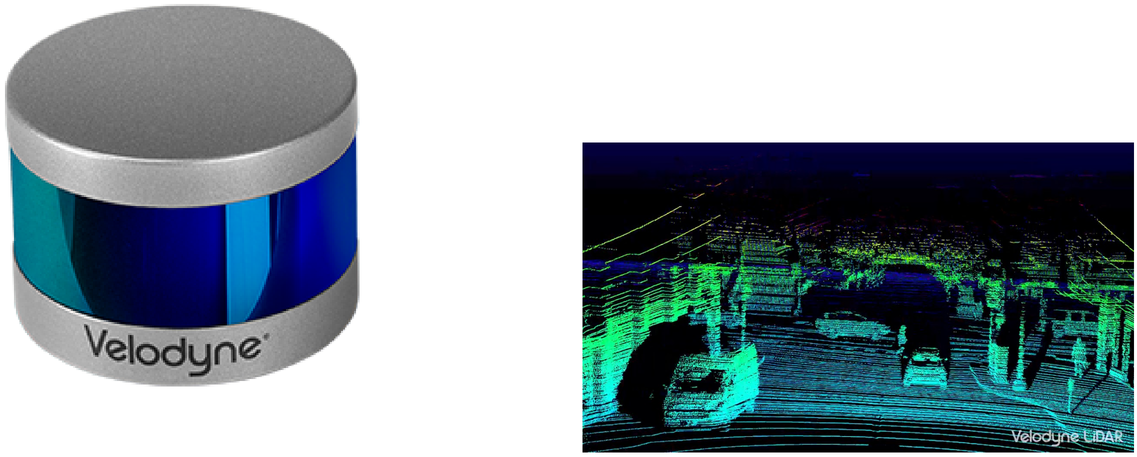


Figure 2: LIDAR sensor Velodyne Puck and data from 3D Lidar

at least something comes back. This becomes a problem when the beam hits either a really unreflective or a really reflective surface. The unreflective surface absorbs all the light that was sent, and the measurement is lost. This can lead to a hole in measurements or even no measurements at all at a given place. An even worse problem occurs when the surface is super reflective such as polished metal, water or even just reflective plastic. The beam can reflect from this material as if it was a mirror and reflect from another different surface. What we get is a false measurement in a higher distance which does not help any effort to localise the robot [7]. The worst enemy for active sensors is water which can be both of those surfaces with even possibility of destroying the robot if it drives into it.

2.2 Non light-based distance sensors

These sensors include other electromagnetic bands such as microwaves and sound such as sonar. Electromagnetic or sound sensors can easily penetrate some objects but have difficulties with others , and it is derived by only material. Those material properties are

(a) GPS, GNSS sensor

(b) Xsens inertial measurement unit sensor

Figure 3: Illustration of non-light based sensors

hard to asses when the environment itself is unknown, but generally, both of those systems can pass fog or other light-obscuring problems. Each of those systems also works on time of flight concept, making them active sensors with usually high energy usage. This makes them unsuitable for long term solutions, and they can break more easily. Currently, sound-based systems are not precise enough since many environment variables cause the sound to travel at different speeds. This includes material (rubble, haze, fog), temperature (fire, snow) or even the flow of material (wind) at given place [8]. Same goes for electromagnetic radars, which have the problem of having to deal with unreflective materials and the problem that for them to perform well use a large amount of outbound radiation. Since most, if not all, countries around the world prohibit usage of strong RF systems by almost everybody except military, it is hard to test and have permission even to use these sensors.

2.3 Direct localisation sensors

These days even the GPS Fig:3a, or any other GNSS for that measure, is considered a sensor even though it just is a receiver for a global lighthouse localisation system. They mostly rely on satellites around the earth which works well in outside conditions without any interference, but in city blocks or even inside buildings, several problems arise. The biggest problem here is just a loss of reception since buildings block RF signals. This causes shortages or even just straight up not having a localisation with this system.

One of the last types is Inertial Measurement Unit (IMU) Fig:3b which can give the robot its speed and orientation in earth coordinates, compasses or magnetometers which give global orientation, gyroscopes that give angular acceleration and barometers which can give elevation estimations. Those units work well with slow-moving objects or when they can be corrected for a drift usually by GPS. However, it can be assumed that inside large buildings, GPS is unavailable. Usually, the presence of metals in surroundings causes the inner compasses and IMUs to sway different direction, causing it to drift or even jump in space. Barometers are susceptible to changes in temperature and humidity and gyroscopes are generally not precise enough to use them as any long term localisation.

2.4 Sensor fusion methods

To improve robots perception of the surrounding world usage of multiple sensors is usually implemented. Usually, it is to either widen the view, fill resolution gaps on needed places, sensing in unusual conditions or just redundancy [9]. Sensory fusion has three main principles [10]. "Redundant" fusion is used when all sensors give the same kind of data improving the system robustness. The system takes data from more sources but uses other higher algorithm that decides which data will be propagated forward. This allows for redundancy and eventually better accuracy since you can select a better sensor for the job or do an average of two sensors lowering the noise effect. This type can be seen as having one distance measuring LIDAR and sonar next to it. In usual operation the LIDAR has better range and better resolution but as soon as the robot entered haze logic switches onto using sonar which is not affected by the haze so much. This results in having at least some imprecise data then no data at all.

"Complementary" fusion can be used when multiple sensors give non redundant data about the world. Using multiple sensors in this way allows for better coverage of one robot. Practical and probably the most basic example is to use multiple cameras on a robot to achieve 360 degrees view of its surroundings. This principle not only increases the data the system can work with but also helps to achieve completeness of all information gathered faster as you do not have to turn around with a robot to scan the whole room.

"Cooperative" fusion uses data from several different types of sensors to improve the gathered data by algorithmic computation. Using sensors in this manner can help achieve results that would be impossible with just one sensor in any condition. As an example, one can see as a stereo camera sensing solution as "Cooperative" where with knowledge about the camera structure, one can calculate distances due to stereoscopic vision. Another example can be the usage of the inertial measurement unit of the system to correct rolling shutter effect on rotating LIDARs by recomputing the actual position of each measurement in the given time with speed from IMU. One can think of this as combining two different sensors to achieve better results.

One can easily just take an average of measured values or just as quickly expand the returned measured values as more measurements. More laborious approaches include probability theory such as Kalman and Particle filter or geometry such as trilateration or triangulation.

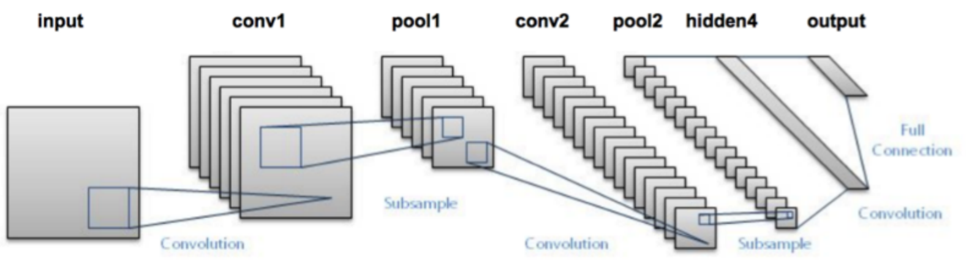


Figure 4: Inner working of a neural network with several convolution layers

2.5 Object detection and localisation

All perceiving sensors give a robot an idea of what is around him, but most of the time is just a lot of unnecessary information. Saving or just sorting all this information is complicated and sometimes downright impossible, so an algorithm that is capable of discerning if the object is interesting has to be developed. Since human objects are usually made to be easily visually distinguished one of the easiest approaches is to make a specialised algorithm to detect the object in camera or use a neural network a teach it how the object looks like [11]. This algorithm takes data from a picture and uses several layers of convolutions to eventually come down to a result which is usually much less data than was on the input Fig:4. This approach works well when the object is well lit, is not covered by any obstructions and is discernable from other similar looking ones in the environment. For everyday use, those three assumptions mostly hold true, but if any of those is false one has to switch to a different domain than light to detect objects. Only possible improvement using sight is to switch to thermal imaging if the object we are looking for is suspected of giving out heat signature. This still usually requires line of sight applications but does not need light.

One such a domain is sound where active objects can be located since microphones can pick up the noise they are making. The biggest problem of this approach is that robots usually produce at least some noise themselves masking distant calling for help unrecognisable in their microphones. Also, the sound is not very effective at transmitting power since 1w speaker cannot be heard behind a closed door. Third and probably most significant domain after the light is the radiofrequency domain where the power of 1w transmission can give you several hundred meters visibility (even-though this power level is banned in most countries since it might cause unwanted effects [12]). Using radio communication is usually the best solution in those applications since it can be transmitted at different frequencies for better accuracy. Radio also penetrates many materials and does not require large reflective surfaces on the object. The biggest problem of usage of RF is the inaccuracy caused by several variables that are tied to the environment it is in. This can be somewhat solved by surveying the environment beforehand, but in situations where you need this localisation, this survey was not either done or just is not valid anymore for example because of cave-in or collapse of the building.

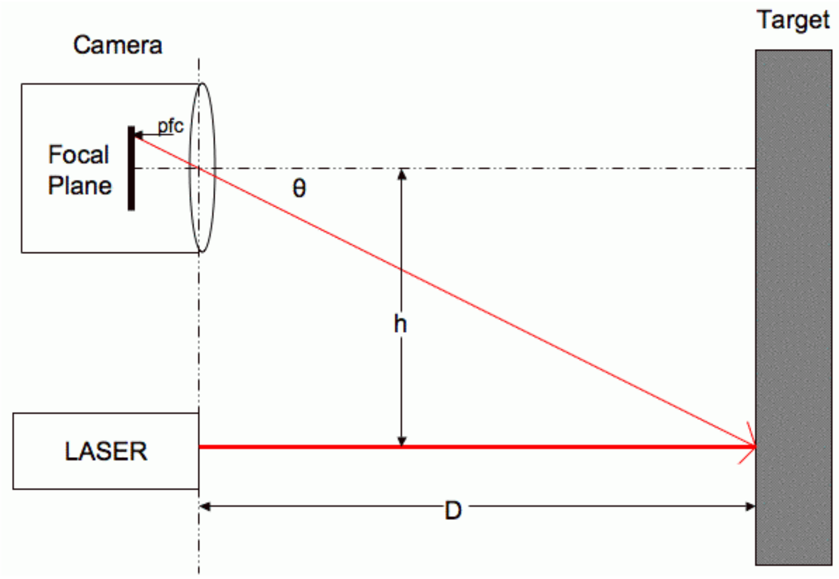


Figure 5: Projection of depth (D) acquired by LIDAR to the camera picture with known position between those two devices

Even though if one has a way of finding an object of interest with a robot, he still needs a way to localise it. This is the exact place where the sensory fusion is essential where the sensors and the robot's position (which is usually also obtained from complex fusion) are fused to calculate the position of the object. If the camera and the visual approach is used the object is usually at the line of sight but projecting a line through the pixels of the camera that found the object does not give a precise location. For this very reason, there can be an additional sensor such as lidar that measures the distances and is able to return measured distance at the place the projected line corresponds to the measurement obtaining the depth Fig:5. Another approach would be to move the robot and then try to make ad-hoc stereo vision using the same camera just from a different position. The latter approach usually does not work since for the proper calculation localisation of the robot is not precise enough.

For sound direction, it is even harder and if the robot does not have a dedicated microphone array that would function the same way as the cameras pixels do. The robot can effectively just play Hot and Cold game using the level of the sound [13]. Not only is this time consuming but also very imprecise since in buildings or caverns with echo the sound levels can be largely distorted.

Radio localisation with directional antennas can produce the same principle as the camera could but without the need of a line of sight [14]. Not only that but also has the same possibility of the signal strength, so the information of distance is also somewhat available albeit imprecise. Suitable property of radio-based localisation is that in any cluttered environment radio is much more uniform than sound-based so a trilateration principle can be applied and used even without explicitly looking for a given object.

To correlate the position of found object a robot needs to localise itself in the world.

2.6 Localisation

Localisation, as it is, uses robots onboard sensors to correlate all those data and to find itself in the world. Using sensory data, one can asses the robot's position relative to some part of the map, its last known position or just the location itself. All of these approaches have their problems. If the robot localises itself only against the last known position ,for example from the distance it thinks it has driven, the error in the measurement grows rapidly due to integrating small bias in each measurement [15]. This problem would be solved with localising itself against the global map, but this approach has its problems. Whenever you either make a map or sense the surroundings with a robot you get small discrepancies. Dealing with imprecise maps or measurements is hard since usually, this gives you more places in map where you can be unless you have some prior knowledge where to look for robots position. Many systems use GPS or other lighthouse based systems for this very reason [16]. Also, if the global map is large, it takes considerable computational power to search all the space for possible matches with your sensory equipment to the point of requiring specialised hardware [17].

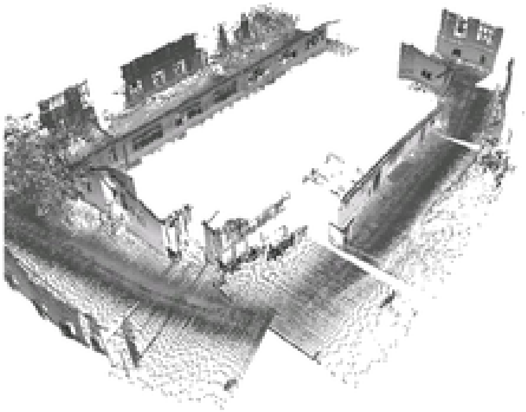
This leads to using all types of localisation at once. The relative from the previous position to get the prior where the robot can be in the global map and then it tries to correct the error of the movement measurement with other algorithms that use the global map. To asses first position in the global map or just a general correction the absolute localisation with GPS or other system is done at the beginning. The last step requires some kind of infrastructure that tells you the position, making it unusable in any environment that is unknown without GPS such as buildings or underground. Thus, without preexisting infrastructure, the robot has to use only its sensors to create the map by itself.

2.7 Mapping

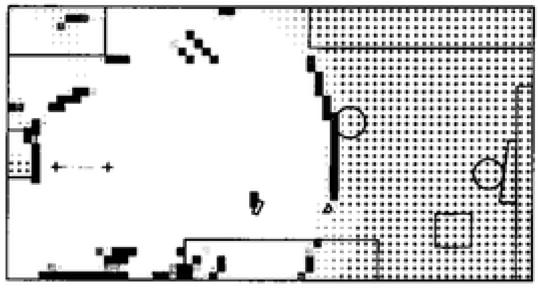
For this purpose, several types of maps are used each with its pros and cons and usually during mapping, more types can be used at once.

2.7.1 Metric maps

The most exact is the continuous metric map which is the exact measurement of surrounding and its real-life dimensions as seen in Fig:6a. This map allows for precise global positioning and measurements in meters (or other exotic units) and has no discrete positions [18]. On top of that are metric grid maps which usually still consider real distances but discretise them onto specific chunks such as a grid [19]. This lowers the size of the data



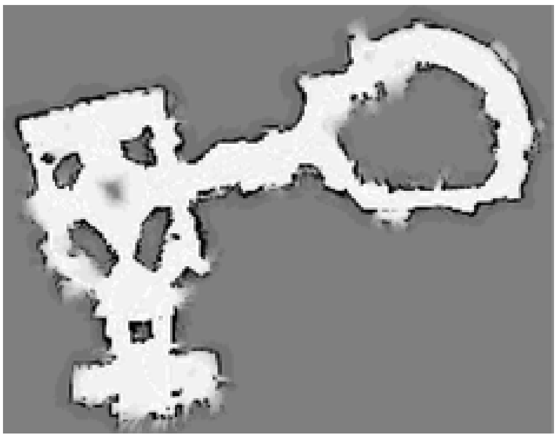
(a) Metric continuous map



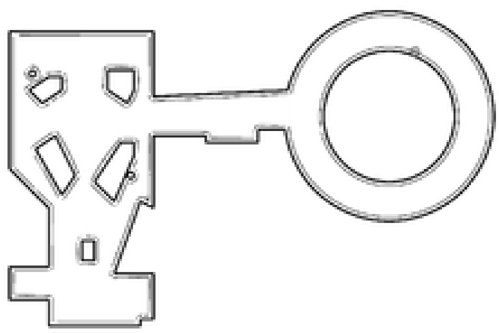
(b) Metric Grid map

Figure 6: Large metric maps

that the map needs to be stored but can lead to rounding errors and other imperfections in the map like Fig:6b. The most common example is a 2D occupancy grid which considers only if each given cell is occupied or not like in Fig:7a. Another examples include 3D occupancy grid is octomap representation where the map is divided into larger chunks and then subdividing chunks with different stored values. This helps to even further save hardware requirements [20]. Improvement on this idea are geometric maps which are comprised of geometric shapes such as lines and circles Fig:7b making them scalable like raster images and less memory demanding then occupancy grids [21].



(a) Metric occupancy grid map

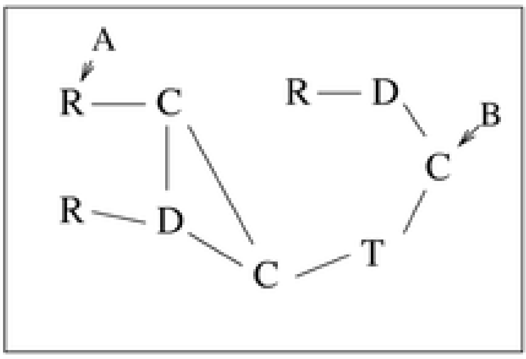


(b) Geometric map

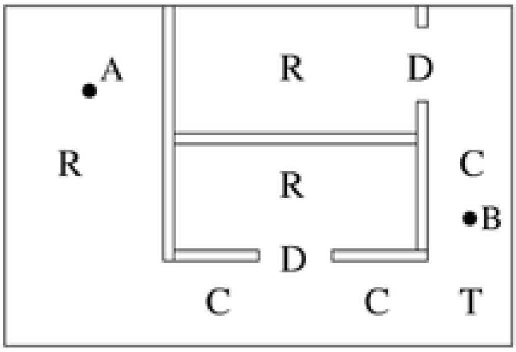
Figure 7: Less data demanding metric maps

2.7.2 Topological maps

Second larger type is topological which does not consider exact distances from objects but rather connects them in a meaningful manner [22]. For example, information that the robot can move from this point in a hallway in the adjacent room but to move to a different room he needs to go down the hallway. Topological maps are effectively graphs that can be easily searched. From a setup like Fig:8a the topological maps looks like Fig:8b



(a) Topological map of the environment on the right



(b) Original template for the topological map

Figure 8: Topological maps

2.7.3 Semantic maps

The last big type are semantic maps which describe properties of some parts with greater detail to allow the robot to localize itself from seeing only one landmark or understanding in which room it currently is Fig:9. The biggest problem of those maps is that the system needs to understand its surroundings on a deeper level such as object recognition or world interactions to implement them [23]. This is basis for landmark-based navigation which is on some level what humans use in real life. As an illustration metric map can be thought of as the classic city map with street names . A topological map can be maps of subway network that only show stations and lines and the semantic map can be seen as tourist map where the significant city landmarks such as tourist places are made much more visible.

Maps are made from all kinds of data, but usually one needs to make the map with the same data it will be used for localisation. To make those maps, one needs to localise himself at the same time, making it a Catch 22.

2.8 SLAM

If a robot goes somewhere where the map is unknown, it would seem that it has to use either some kind of lighthouses or just the relative localisation. For this very reason,

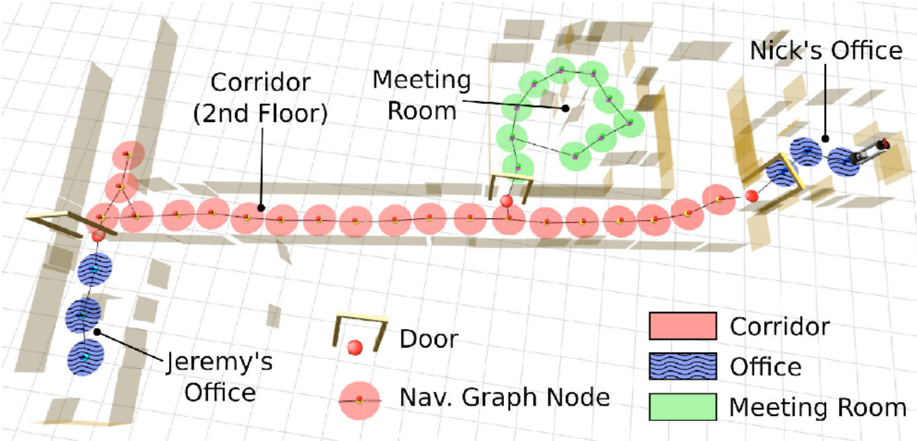


Figure 9: Semantic map with a vague description of a rooms

several Simultaneous Localisation And Mapping (SLAM) algorithms have been developed. There are several types and approaches to SLAM where they can work using cameras, lidar, ultrasonic sensors, GPS, and any other sensors to estimate the position. These approaches include but are not limited to Kalman filtering [24], particle filtering [25], iterative closest points [26], visual slams [27] or any combinations of the above. This makes them a prime example for sensor fusion since one needs to usually use multiple sensors to make the map. Most of those algorithms run in a loop where they try to make a local map from given sensory data, then when robot moves they try to correct the movement measurement by matching the previously seen part of the map onto sensory input, and then they make a new part of the map from new data the sensors see now.

Some can propagate all previously measured data if a robot comes back to a known position known as loop closure [28]. Some are capable of being able to localise robot without any prior knowledge of his position in map called a kidnapped robot. Some will adjust its map to dynamic objects or seasonal changes or even predict changes in surroundings based on current knowledge [29]. Describing all of those is out of the scope of this work.

2.9 Quality assessment

All approaches described above have multiple possible implementations, so one has to choose one. Most of the times, this is derived from what sensors and what is the intended application, but more often than not, one has to make a choice from several algorithms. To compare, choose and eventually make better algorithms, one needs to know which are better than others. This requires to have some kind of a metric that is able to evaluate the performance of said types of algorithms. Usually, it means quantifying some kind of a metric such as an error, precision, usage or any other values that correspond to the quality of the algorithm. In some cases, word description is enough but the data usually still needs to be compared easily or displayed in a graph calling for numerical evaluation of the quality. Generally, those values can include hardware and computational resources needed to compute wanted results sufficiently. Moreover, each type of task usually has its own key performance indicators (KPI).

2.9.1 Detection

For detection problem one is usually interested if the system is able to say that the object is in the vicinity and if this estimation was correct. For this reason, several techniques are used out of which almost all use paradigm of true/false positive/negative. True positive (TP) detection is when system estimates that it has detected the object and the object is there. This is directly contrary to false positive (FP) which means that system guessed badly and the object is actually not there. For negatives is almost the same, as in a true negative (TN) means that system did not detect anything and nothing was present, but false negative (FN) means that the system had missed the object.

With these four values, the perfect system has no FN and FP making it detect all possible objects it could have while simultaneously not making detections where it should not.

Those four values create confusion matrix allowing for much more different computations possible to show different results. Most are out of the scope of this work. Firstly the receiver operating characteristic (ROC) curve is the most sought after if the distribution of actual positive and negative values are somewhat the same [30]. This curve is a ratio between TP rate (TPR) and FP rate (FPR) that can be calculated as

$$TPR = \frac{TP}{TP + FN},\ FPR = \frac{FP}{TN + FP}. \tag{1}$$

This curve can show how separable are the two instances for detection, making the area under the curve directly proportional to how good the detection system is. This has a problem in an environment where the actual instances of the wanted object are unbalanced and are far less in numbers than all possible places they can be at. To evaluate systems like these, precision and recall curve is used since it correlates the accuracy of the estimation to the total points guessed during the detection.

To make such a graph one needs to calculate recall which is the same as TPR and precision which is calculated as

$$precision = \frac{TP}{TP + FP}. \tag{2}$$

It means that both precision and recalls are solely estimating the positive parts and ignoring the true negatives [31].

Both of those methods require the system to have a quantifiable amount of negatives and positives. If that statement is not true one needs to use different evaluation method. However, if the amount of objects that one has to detect is constant, it is possible to show what percentage was detected. This approach has only one problem where if the system says it has detected everything this value will always be 100%. Usually, in scenarios where there is a stable number of objects, it is also limited how many tries the system has, which should be mentioned with the percentage results. This ensures that the value will not be 100% all the time and can be helpful to evaluate the system.

2.9.2 Localization

In the localisation problem, the most sought after the indicator is how precisely is the system able to determine the position of the object its suppose to find.

This error is usually denoted as

$$e_e(m) = \sqrt{(x_x - m_x)^2 + (x_y - m_y)^2 + (x_z - m_z)^2} \tag{3}$$

or as

$$e_m(m) = (x_x - m_{ix}) + (x_y - m_{iy}) + (x_z - m_{iz}) \tag{4}$$

where each m corresponds to an estimated position and x to ground truth position in one of the coordinates and e to the "error" of the estimated position. This first equation corresponds to Euclidean e_e distance which calculates the shortest path from the ground truth to the estimated position [32]. The second equation is a representation of Manhattan e_m distance, where the total sum of errors is used. Manhattan distance is useful if the world is discretised to grid-like structure or then the robot is only able to move in directions with 90 deg increments. This situation can occur, for example in long hallways with small rooms to the sides or in grid-like search [33].

To evaluate the error, one can calculate if from just the final estimate after a set amount of time which should be plentifully to converge. This allows comparing the system by only its final verdict it gives about the position it had to find. Better solution is to obtain "mean error" which can be calculated as

$$\overline{e(m_1, m_2 \dots m_n)} = \frac{1}{n} \sum_{i=0}^{n} e(m_i) \tag{5}$$

where n is the total count of measurements.

This value shows how imprecise was the localisation throughout all estimates [34]. One counterargument against this is that the system that had some large errors with later small errors will have the same value as a system with average values throughout the whole measurement. To accommodate this problem the root mean square error (RMSE) [35] is used which is defined as

$$rmse = \sqrt{\sum_{i=0}^{n} \frac{e(m_i)^2}{n}}. \tag{6}$$

This allows us to give larger errors a more significant impact on the final average.

2.9.3 Mapping

Since maps are much more complex then detections which are true or false and localisation which is just a position, one needs to use quite a sophisticated algorithm to evaluate it. With no set standard, many approaches are used. In [36], the entropy estimation is made and basically compared how many environmental data the map uses. Alternatively, whole disaster city testing site [37] was made to evaluate multiple aspects of maps which include metric precision of each point, a structural description of graphs, data size requirements, computational resources need and few other metrics but describing those are out of the scope of this work, and the interested reader can see [38].

2.9.4 SLAM

Evaluation of slam algorithm calls for a need of ground truth acquisition to compare the true position of the robot to the world frame. This is usually done using external localisation system such as RTK-GPS which gives the precise position of the object or markers which give the position of the robot towards the detection device. For this purpose geodetic total stations can be used with submillimeter precision but are expensive. A different solution is to use fiducial markers which only require a camera and still give precision to within 5cm [39]. Aside from the price fiducial based systems are also usually capable of localisation of multiple robots at once making them useful for robotic teams.

Simultaneous localisation and mapping precision are as complicated as mapping performance evaluation, but two main metrics are universally used in recent research [40]. The first one is relative pose error (RPE) which measures local accuracy over a short period of time which shows if the SLAM is drifting. This is a useful metric in one uses odometry systems since they are prone to constant drifting. Another used metric is absolute trajectory error (ATE) which basically just calculates distances of position estimations from ground truth in some sort of a map. Several other metrics such as computational resources, robustness to kidnapped robot problem, closed-loop solution are also used often but are out of the scope of this work.

3 DARPA SubT

Since the goal of this work is to "Design and implement a multi-sensory system to detect and localise objects transmitting radio, chemical, and audiovisual signals in adverse conditions of disaster scenarios." it was also meant to be implemented and tested on the system used on DARPA SubT competition by CTU.

DARPA's Subterranean Challenge (SubT) is one of the contests organised by the DARPA to test and push the limits of current technology. In this competition the task is to find and localise given objects with 5m accuracy within a course spanning up to 8km [1]. This contest is done on several different underground courses with unknown layout or structure. During each round robots are expected to be operational in conditions such as mud, fog, rubble, deep and dripping water, light/dark shifts, steep inclines and dynamic obstacles while being supervised by only one operator sitting outside the course.

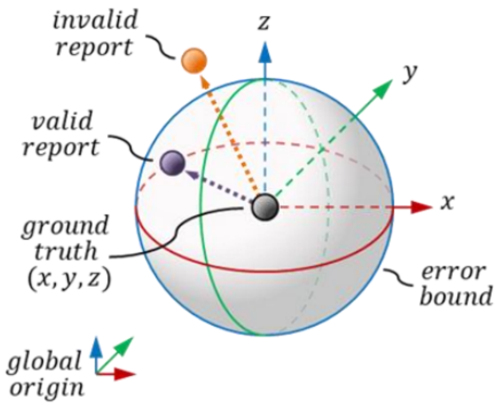
3.1 Runs, environment and scoring

Four separate rounds of the SubT challenge each occurring at different place and time happen in different environments with simulated conditions. Basic information about the type of the course is known beforehand but is vague such as "Tunnels, Urban, Cave" environment. "Tunnel" round was held in NIOSH Experimental mine in Pittsburgh which was resembling man-made tunnels, and "Urban" was held at Satsop Nuclear power-plant in Washington state. Two of those rounds already happened since the competition is held from September 2018 until August 2021 Table:1.

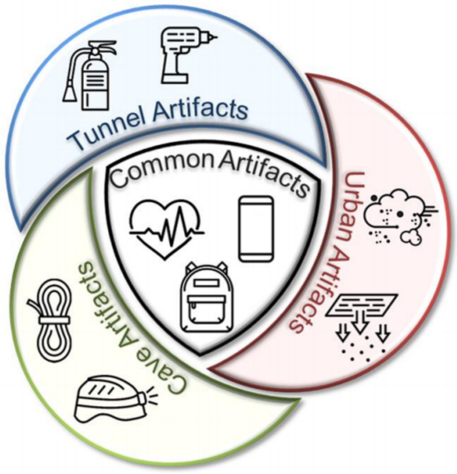
Table 1: Timeline of the competition [1]

Event	Date
SubT Integration Exercise (STIX)	April 2019
Tunnel Circuit	August 2019
Urban Circuit	February 2020
Cave Circuit	August 2020
Final Exercise	April 2021
Final Event	August 2021

In each of this competition the organizers give out information about what the objects of interest are with their precise specifications and localisation points calling those "artifacts". So far each round had two tracks to accommodate runs of multiple teams at once. Teams are given one hour on a track to localise and report proper position and type of objects given beforehand. If the location is within 5m to ground truth as in Fig:10a and type of artefact reported to the DARPA team is correct team gains one point while losing a point



(a) Precision needed for scoring



(b) Artifact list for each round

Figure 10: Visualization of rules [1]

if the same artefact is reported the second time. Each team has two runs on each track, and then its final score is a sum of the best runs from each track. This means that whoever is able to find the artifacts faster with greater precision gets the lead.

3.2 Artifacts and detection

In the first three round of the competition, only five artefact types are used while the final round is supposed to have all types combined Fig:10b. Some artefacts are constant throughout all the rounds such as Survivor, Backpack and Cell phone [41]. While localising given object, one can find it by sight, sound, olfaction (smell, small particles dissipating through the environment) and radio footprint. While using heat signature can be technically considered sight with the need for a line of sight, it does not require light to detect the object, so it is considered different domain later on. Some of these artefacts can only be detected, and localised using light others have different properties which can be abused to at least give a sense of direction to them or straight up give a location without the need of a line of sight to them Table:2. Currently, only one artefact was special that it was not detectable by light at all. This forced us to implement at least one other domain to the detection process.

Since artefact detection from an image was done using a custom trained YOLOv3 neural network omnidirectional camera, this work mostly focuses on all other domains detectable using the same platforms. Every detection that is done with cameras is sent to the human operator whose job is to command robots and confirm the artefacts to be sent to DARPA.

Table 2: Artifact list and usable domains [41]

Artifact	Rounds	Domains
Survivor	All	Light, Heat, Sound
Cell phone	All	Light, RF, Sound
Backpack	All	Light
Fire extinguisher	Tunnel	Light
Acu Drill	Tunnel	Light
Air went	Urban	Light, Heat
Elevated CO_2	Urban	Olfaction
Climbing rope	Cave	Light
Climbing helmet	Cave	Light

3.3 Key performance indicator requirements

Since the goal of the competition is to precisely detect, localise and report back the position of sought after artefacts algorithms need to be evaluated with those requirements in mind. The detection of the artefacts can be evaluated, such as described in 2.9.1 from which the last metrics make the most sense. It is used mostly due to information that there is always a stable amount of instances of artefact type on the track, and the total amount of guesses that one has to pinpoint them is limited. For the gas artefact, the amount was three and for phones four. This information forces us to limit the number of guesses for phone artefact type to four since it is the amount that would be used during the run. This also incorporates the idea that the phone artefacts can also be detected by visual means, and this leaves half of the available guesses to it. The same paradigm also corresponds to CO_2 detection, where the limit is six since they cannot be detected by any other means.

For the object localisation, the minimum error might also be helpful since once we already estimate the position precisely enough we do not have to seek it more precisely anymore because the point had already been scored. Another useful metric for this situation is calculating average error and RMSE since each describes a little different information. The average tells us that if we have not reached the necessary precision to score a point how close we have been most of the times. The RMSE will show how much we have missed by a large margin since it will be bigger than average the estimations had some really bad errors which are even less useful then semi-precise estimates.

Another metric that we wanted to use was to include data “Amount” that is required to find the object. This metric is not connected to the actual guesses that would take place since this just measures how much data was needed for the correct position estimation.

Fewer data needed for the correct localisation is helpful to form many perspectives. For example robot with less needed data “Amount” for one place has to store or send back to base, less of the data making it cheaper or sometimes even possible to build such a system. Since the data is collected on some timely basis, it directly correlates with the speed it can be collected with allowing for shorter searches to be performed, the less data is needed. This principle shortens the time of area being searched, making it again less pricey to implement the system. Having to spend less time in the environment also helps in a situation where the environment is somewhat hostile to the platform, such as corrosive or radioactive elements present in the surrounding. This again makes it more viable to deploy any search system if it does not need to be destroyed or severely damaged in the process. As seen in the algorithm 1 to compute "Amount" one needs an acceptable error. It corresponds to a level of precision with which the system needs to perform.

Algorithm 1: Computation of needed amount of data

Result: Amount of needed measurements to achieve given precision c
measurements M of size n, ground truth X, acceptable error e ;
for $i \leftarrow 1$ **to** n **do**
  $E = e_e(m_1, m_2 \dots m_i)$;
  if $E \leq e$ **then**
    c = i;
    return c;
  else if $i = n$ **then**
    c = NaN;
    return c;
  end
end

3.4 Robots and platforms

Platforms that were used comprised of three old BlueBotics Absolem tracked robots, one wheeled Husky A-200 Fig:12 and two quadcopters [42]. All other parts necessary for

Table 3: Used key performance indicators to evaluate the system

Name	Short	Unit	Ref
Smallest error	E	meters	(3) (4)
Average error	$\overline{E}$	meters	(5)
RMSE	$RMSE$	meters	(6)
Data amount	C	count	1
Detection percentage	P_{Wi-Fi}	% / estimates	2.9.1
CO_2 detection percentage	P_{CO_2}	% / estimates	2.9.1

the functioning robot fleet supervised and controlled by the human operator were solved previously by the CTU team participating at SubT using Robot Operation System. Each of the big robots was previously equipped with an omnidirectional camera, 3D or rotating 2D lidar, Jetson TX2 and second computer wither NUC8i7 or equivalent. Those prerequisites meant to implement the system to already living environment a forced a lot of constraints and design decisions that will be described later. Constrains being that the system has to work within the unknown environment, without hampering the functionality of the already given system while simultaneously be robust to loss of communication of robot with a base or with robots that lost its localisation. During a run, if communication between robots and the base is lost, robots store their information in a database that is sent back to base as soon as the robots return to a signal.

Robots were carrying several types of communication modules from only two types were used during the actual run. One type of communication was using Mobilicom MCU-30 Fig:11a units which are out of the box mesh capable devices which utilise 4G technology at 2,4Ghz and time multiplex. With this communication, each robot has a dedicated time slot where it can be transmitting or have its signal retransmitted by surrounding units. This mechanism limits maximum bandwidth of system to about 300KBit/sec per each robot while requiring a rather large unit to be carried by the robot itself. Only three tracked and one wheeled robot had this unit attached which allows for video feed as well as better control and information about the system and its surroundings. Contrary to previous so-

(a) Mobilicom MCU-30 communication unit

(b) MOTE communication unit

Figure 11: Communication modules used in DARPA SubT

lution smaller communication modules called Motes Fig:11b were attached to each robot in the fleet. Their main appeal is usage of 900Mhz band which much better penetrates surroundings and their small size of 5 by 5 cm PCB. Price of 20 USD (compared to several hundred USD MCU-30) per module allowed for them to be deployed throughout the run by larger bots to cover as much track as possible with coverage. Limiting factor is the bandwidth of this network which is about 60B/s since this system uses flood network. Small size allow small robots such as hexapods and UAVs to carry and utilize those communication modules [43].

During a run, it was preferred to use Mobilicom with its higher bandwidth where pictures of detected objects could be sent to a human operator for clarification. However, when robot lost connection to the base with Mobilicom channel, the necessary data such as object detection without the picture or commands to robots were sent using Mote network.

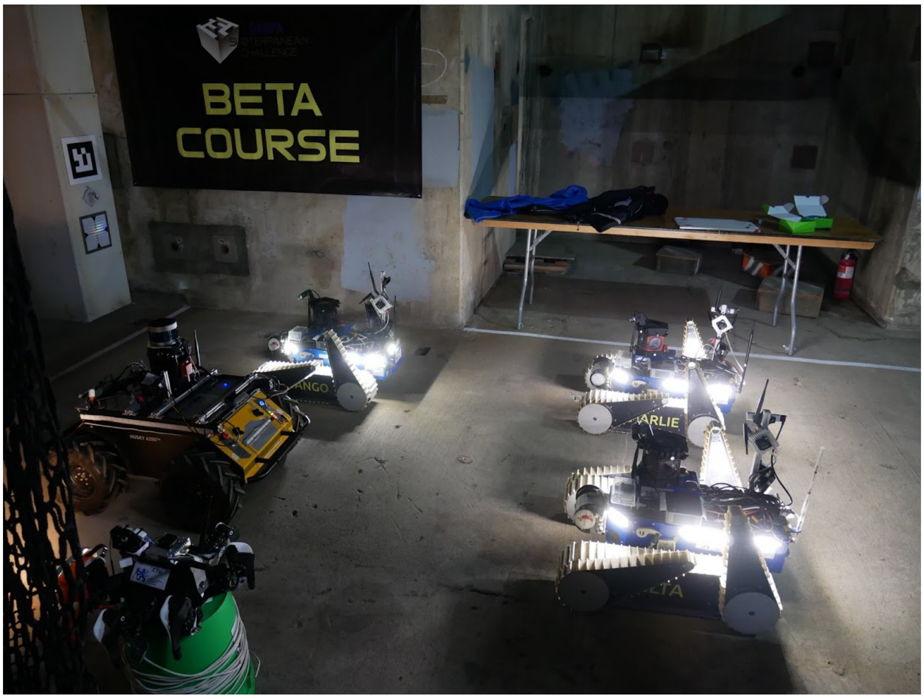


Figure 12: Fleet of CTU-CRAS-NORLAB robots during DARPA Subt Urban round

4 System description

4.1 Approaches

Several approaches were tried with various levels of success. The decision was made to try to focus on domains that can help us to perceive in darkness, fog or even through walls. This mostly left all of the optical approaches out of the equation which can be explored later on in future work.

The first approach was to implement olfaction using CO_2 detection sensor. This worked well enough with only minor problems such as that the sensor needs to be calibrated since it is temperature and humidity sensitive.

Another approach that was experimented with was the usage of microphones to leverage the sound the first two artefacts are supposed to make. Mostly for the reason that three of the four robots already had a microphone on them and because those two artefacts are supposed to be present at all rounds. Any tests with microphones showed that for a robot to be able to hear a speaker of a phone at full volume, it needed to be less than 5m from it while simultaneously being stationary. This approach worked only for the wheeled robot, which while stationary has little background noise such as fans or other moving parts. Since all of the other robots use some kind of perpetually moving parts, the idea was scrapped.

Next domain that allows for an extended reach around corners was the usage of Radio Frequency. This was encouraged by the idea that the current artefact detection did not work as good for the Phone artefact as it did for the rest. This was mostly due to changing conditions in the environments which made all the phone screens white rectangle making it unrecognisable from any kind of distant light source or just a simple paper on a ground. Since in this competition it was possible to capture Wi-Fi and Bluetooth signals for the artefact this was said to be used as a possible improvement to the system.

4.2 Radio detection

Usage of Wi-Fi and Bluetooth (BT) for localisation is not a novel idea [44]. Additionally RFID tags are also quite popular for localisation [45]. Several possible approaches are possible, but all of them require scraping of the Wi-Fi and BT data. Wi-Fi works on making an Access Point (AP) which then usually acts as a master and allow other devices to connect to it. This usually means that one needs to be able to see the AP while scanning surroundings to know where it can connect. While the AP is scanned, it tells the client its SSID (Network Name) and MAC address.

The client is also able to measure the Received signal strength indication (RSSI) value that corresponds to signal strength [46]. Since we can measure the RSSI of each device and we know our position in the map, we can drive around and calculate from our previous positions and signal strength the position of the AP. This requires the AP to be stationary, but in our case, it satisfies this condition. RSSI is also highly non-linear a noisy, which makes it hard to extract data from it.

4.2.1 Distance

RSSI vaguely corresponds to the distance to the object you are on a path to find since it drops with the distance to the radio[47]. It is a measurement of power received by radio usually ranging from 0 dBm (Perfect signal) to -100, which is an extremely bad signal. With usage of

$$r = 10^{\frac{a-rssi}{10\cdot n}}, \tag{7}$$

where $r[m]$ is distance of a radio, $a[dBm]$ is signal at 1m distance, $rssi[dBm]$ is measured signal and $n[-]$ is path loss exponent for given environment, we can calculate the distance [48].

Constant a is acquired experimentally for each configuration on the robot by measuring it but n is a different story.

Table 4: Path loss exponent (propagation constant) examples [49]

Environment	n
Free Space	2
Urban Area	$2.7 - 4$
Indoor LOS	$1.6 - 1.8$
Indoor general	$4 - 6$
Indoor factories	$2 - 3$
Indoor through floor	$3 - 6$

Path loss exponent is a constant that describes how the environment is penetrable, or it allows for reflection of RF. To asses this value, one must either make an educated guess or measure the value in the environment from previously gathered data. Furthermore, this value changes depending on the frequency the system uses, which eventually allows to asses the environment using multiple frequencies at once for greater accuracy if multiple frequencies are in use. In an urban environment such as underground stations, this value is usually between 2.2 and 3 for 2.4 GHz, which is used [49]. In the competition runs the value was calibrated on previous runs in that building.

Measurement of signal level is plagued with heavy noise since we do not know the orientation of the transmitting antenna and the environment topology. This noise can span dozen dBm in the measured data, making distance conversion more imprecise at larger distances since single dBm has a much higher impact in a low signal scenario. A solution to this problem is to guesstimate the precision of the measurement and incorporate it into the localisation algorithm. Educated guess can be made by observing the standard deviation of the power levels at a constant distance. With this assumption, it can be said that the stronger the signal is, the more precise the measurement is. As an example signal can be -40 dBm at 5 meters with a line of sight to the device but if there is a wall the signal can drop to -75 dBm, but it is not like this another way around. This means we can give measurements with low signal smaller weight, or we can narrow down the uncertainty of their distance.

4.2.2 Object location map

One possible approach is to use a map of object location. We begin with a discreet map of probable positions for each object we want to locate. This map can have a uniform probability or can have prior knowledge incorporated. Then for each measurement, we acquire its position and draw a circle at this point with a diameter corresponding to the acquired distance. Then the Gaussian blurring is applied with the size of the kernel corresponding to approximate uncertainty and make intermediate circles that mark an estimated position of the object. This creates a "mask" which corresponds to a position of an object in a map for this very measurement. We then multiply this mask with an object map for a given object. This gives us an updated object map for the next measurement Fig:13.

This solution not only gives one estimation of the sought after position but multiple and can show a general direction in which the object might be located. The most significant problem is the memory usage of a map which can use several GB of RAM with larger (100m side) maps. Extension to 3D proves to be even more difficult since the Gaussian blur needs to be done for each dimension. This requires to have a separable kernel of the blur which limits its capabilities and is not implemented in any commonly used libraries.

A working prototype of the 2D probability map had been made and can handle multiple layers which make it possible to collect data for all object that we want to localise. First 3D attempt with functional blurring and multiplication was made but was never refined enough since the focus was shifted towards more human-readable solution [50].

4.2.3 Lateration

Since the operator required only one most probable place for each phone in range to reduce clutter on his screen lateration approach could be used. Lateration works by measuring the distance to the object from known positions and then using those positions to asses the location of the object. In 2D settings, only three measurements on different positions are required to obtain the object location, but in 3D the minimum amount rises to 4. One can imagine that each distance measurement draws a sphere around a measurement point, and with enough sphere, it is possible to compute their intersection Fig:14. Since

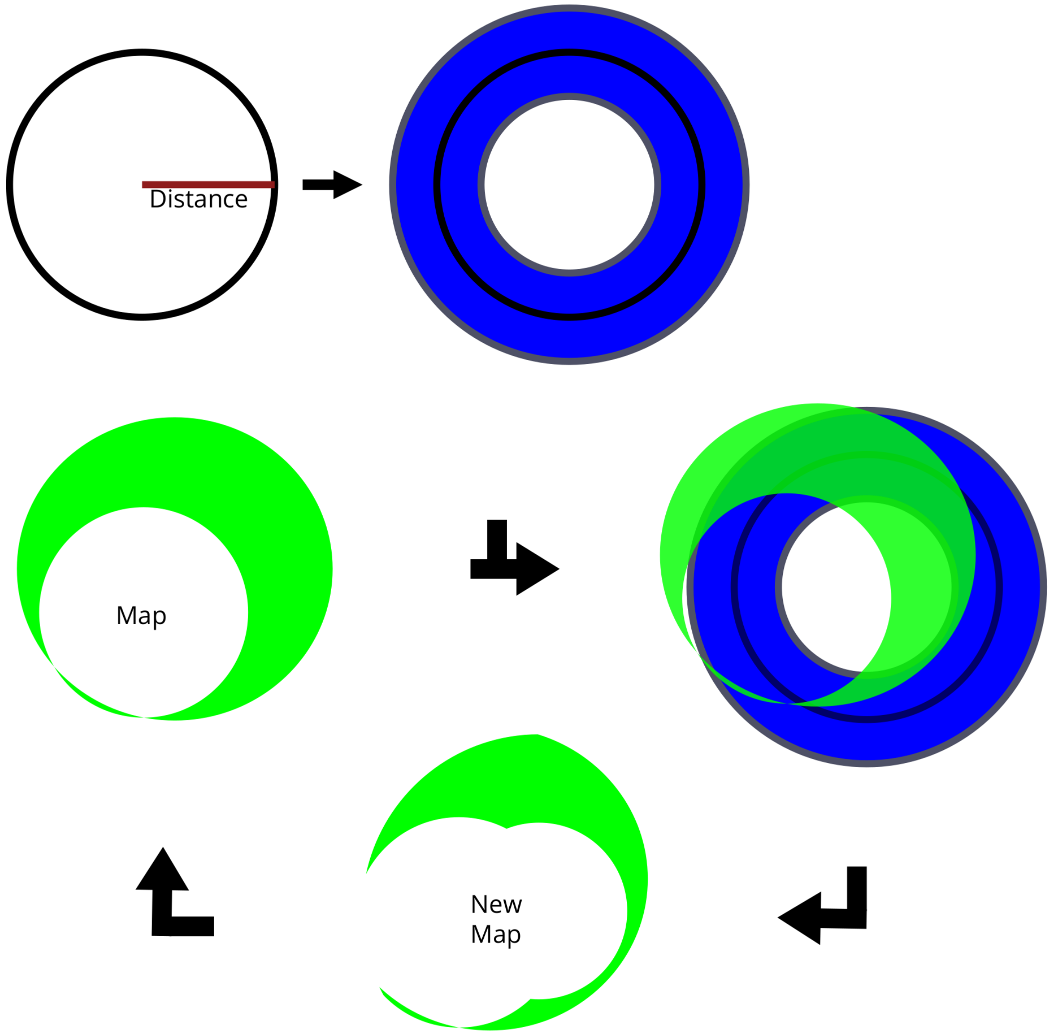


Figure 13: Object map principle

our distance is highly imprecise usage of multiple points in the equation helps to asses the correct point much better. This principle is called multilateration [51]. Sphere equation

$$r_N = \sqrt{(x_N - x)^2 + (y_N - y)^2 + (z_N - z)^2} \tag{8}$$

where $r_N[m]$ is the diameter of the sphere, $x, y, z[m]$ are coordinates of the object we want to locate, and $x_N, y_N, z_N[m]$ are positions where we took the measurements from can be used to calculate coordinates of the wanted object.

Acquiring N measurements, we can rewrite this equation into

$$\begin{bmatrix} r_1^2 - r_2^2 + x_2^2 + y_2 + z_2 - x_1^2 - y_1^2 - z_1^2 \\ r_1^2 - r_3^2 + x_3^2 + y_3 + z_3 - x_1^2 - y_1^2 - z_1^2 \\ \vdots \\ r_1^2 - r_N^2 + x_N^2 + y_N + z_N - x_1^2 - y_1^2 - z_1^2 \end{bmatrix} = \begin{bmatrix} 2 \cdot (x_2 - x_1) & 2 \cdot (y_2 - y_1) & 2 \cdot (z_2 - z_1) \\ 2 \cdot (x_3 - x_1) & 2 \cdot (y_3 - y_1) & 2 \cdot (z_3 - z_1) \\ & \vdots & \\ 2 \cdot (x_N - x_1) & 2 \cdot (y_N - y_1) & 2 \cdot (z_N - z_1) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{9}$$

where the larger matrices can be expressed as

$$\mathbf{a} = \mathbf{B} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{10}$$

. Given previous matrices one can compute the final coordinates as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{a} \tag{11}$$

which effetively calcualtes least squeres method [52]. Advantage of this method is that it does not need more memory what it needs to store the measurement data making it lightweight for the system [48].

Moreover, this solution works natively in 3D without any more problems.

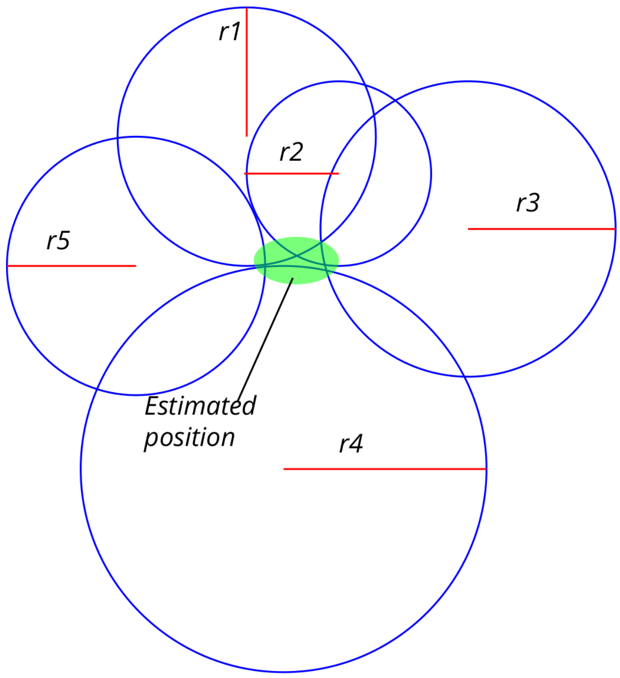


Figure 14: Multilateration approach principle with 5 measurements

4.2.4 Chunking and weighing

To ensure that measurements are taken from different places, all measurements locations are discretised onto 10cm chunks. If a new measurement is made and it belongs into an already measured chunk, it is then averaged with all the previous measurements from that chunk. This means our N in all equations is actually a total number of visited chunks where r_N is an average of its own measurements and x_N, y_N, z_N are coordinates of the centre of that chunk.

This is incorporated together with weighting based on the distance mentioned in the previous chapter. Modifying 10 to

$$diag(\omega)\mathbf{a} = \mathbf{B} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{12}$$

where ω represents weights for each chunk. This is calculated as

$$\omega = \begin{pmatrix} \frac{\sqrt{c_2}}{r_2} \\ \frac{\sqrt{c_3}}{r_3} \\ \vdots \\ \frac{\sqrt{c_N}}{r_N} \end{pmatrix} \tag{13}$$

where $c_n[-]$ is a measurement count for each chunk and r_N is a average distance of the given chunk. This weighting makes the algorithm to incorporate the precision measurement from given chunk as well as the distance weighing [51]. Using square root, we ensure that when the robot stands on one chunk for a long time, the measurements get diminishing returns from each new measurement.

4.2.5 Implementation

Implementation of the system mentioned above was done mostly for the purpose of DARPA's SubT challenge on CTU-CRAS team robots. This required choice of sensors able to scrape the needed signal from phone Samsung Galaxy J8 J819M/DS [41] provided by DARPA. All of the robots had onboard Wi-Fi, but most of them only had 5 GHz modules making it useless since the J8 only has 2.4 GHz Wi-Fi. For this reason, NVIDIA Jetson TX2 with Orbitty carrier board was selected as a sensor of choice since it is capable of 2.4 and 5 GHz Wi-Fi as well as Bluetooth and was already present on all larger platforms in the CTU-CRAS. Jetson TX2 also allows for the use of only one antenna since it uses Bluetooth and 2.4 Wi-Fi on the same connector. This allowed for the usage of only one omnidirectional antenna attached to the housing of the Jetson TX2 normalising the calibration process for all robots. Due to rules of the competition the phone would not respond to any requests of pairing or connection making it rather hard to acquire RSSI data.

To do so, custom scraping software had to be implemented. For the Bluetooth, the only way how to access the data was to use built-in C libraries which sadly did not allow for fast scraping if the device was not paired. This resulted in acquiring the signal strength only once per 30 seconds and in testing showed large inconsistencies if the devices moved in that time frame, making it unusable for this application.

Using Wi-Fi allowed to obtain RSSI, BSSID (network name) and MAC address (unique device identifier) to be collected with a 1-second interval which was consequently sent as a custom topic to ROS running on the robot. This was done for Wi-Fi using C++ which required having a network card where the drivers can force refresh often enough. Luckily this was the case on Jetson TX2, but from testing, we have limited success even on the same brands and model numbers of devices since they may use different network cards. The most significant problem if this system is that it heavily cripples maximum speed achievable by the Wi-Fi module since it forces refresh every second and scrapes all available channels. In testing the top speed of Jetsons Wi-Fi decreased from 50Mbit/s to about 5Mbit/s tho keep in mind that this was done using only 2.4 GHz antenna and without the utilisation of proper 802.11/ac standard.

All scraped data were then filtered to match specific pattern in BSSID using regex expression in the SubT rules stated that all phones would have BSSID as "PhoneArtifactXX" where "XX" were two randomly selected characters or numbers from ASCII table [41]. All this data for each of the large robots were then sent trough high bandwidth MOBILICOM network to base station where they were combined using methods mentioned above.

The first method that was only tested at the beginning was the object maps which were made using ROS grid_map package. This allowed for easy stacking of multiple detected artefacts on top of each other. This method proved to be working well on small scale planar environment (map size of 100m each side) but almost unusable on large scale maps (up to 1Km per side with several floors) required by the competition. This was mainly due to the need for convolution over the whole map with each direction making it not only RAM demanding but CPU heavy.

This forced usage of the multilateration method, which was with minor optimisations able to achieve unlimited sized maps since it did not store nor compute any grid map. With the use of C++ and implementation in ROS system was able to conserve a lot of memory requiring 0.4 GB for a clean run and accumulating up to 0.8 GB after 3600 measurements (1 hour). Since the whole system of 4 robots with scraping enabled this allows for data to be computed from each independently making four separate and one joint instance of code running where all the data are combined together to fit under 5 GB of necessary RAM making it runnable even on low specs machines with 8GB of memory. This is done since it is not possible to just combine all robot data since only one which would have bad localisation could skew the whole result. In practice as soon as the operator designated that one robot had its localisation broken the system switched to all separate instances without joining the data.

In future work, it might be possible to weed out the unwanted data from one robot

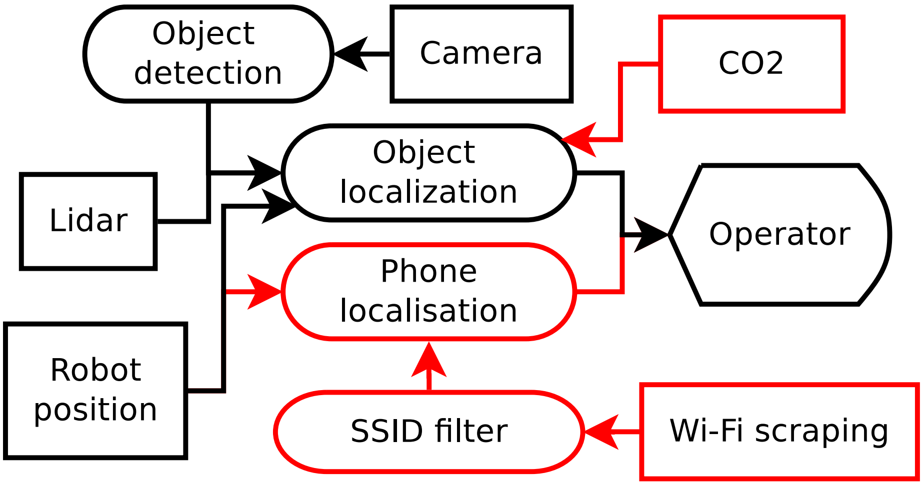


Figure 15: Additions made in this work (in red) to preexsiting detection system

dynamically, and the rest have still connected. This might be possible with a little more powerful CPU since currently with 4000 measurements Intel i7-8th gen @ 2.2 GHz is able to compute the calculations in 360-433 ms and with 16000 measurements (4 robots for 80 minutes) it takes 1.4-1.9 seconds, making it not updatable with each data point.

Since the multilateration returns one position of the object, it is then sent to the operator as a detection which is visible on his screen. This detection includes the position where it is detected and information for the operator how good the estimation is. To display the precision estimate is made based on the closest measurement since we already know that the closer one is, the more precise the estimation is.

All parts Wi-Fi scraping, SSID filtering, CO_2 detection and phone localisation was added to preexisting pipeline used by the CTU-CRAS-NORLAB team in DARPA SubT Challenge showed in Fig:15.

4.3 Olfaction

System for the detection of CO_2 was implemented using the recommended sensor SCD30 from Sensorion Fig:16. This sensor is capable of CO_2 concentration measurement range is $400 - 10000 \pm (30 + 3\%)$ ppm. The sensor is also capable of measuring temperature and humidity, helping it to measure more accurately in different scenarios. This sensor was installed on all tracked wheeled and areal robots using I2C interface to Arduino Nano which reads the data from the sensor and makes it ROS compatible message. Arduino Nano is then connected via USB to Onboard PC which gives it a timestamp and saves the data to the database which is eventually sent back to the base station using an appropriate channel. With the CO_2 level data were measured each second and then compared to the measured value at a given robot position. If the data on the given position were higher than previously measured ones, then the value location was updated to the highest measured

Figure 16: CO_2 sensor SDC30 from Sensorion

value at the position. This was done because the sensor has 20 seconds response time on rising edge before it accurately measures the increase of co2 in surroundings. Measuring higher value in given position was better for not missing possible places with increased CO_2 levels. If the level of CO_2 was higher, then given threshold specified by the rules the detection was created, which includes just one measurement and its position. This helps the operator to eventually turn off all the measured points and just look for threshold places with high concentration since only one detection can be created in a 5m to ease the cognitive load of the operator. All measured Co2 level data are sent back only on the high bandwidth network while the singular detection can be sent using the MOTE network.

This implementation allowed all robots to be able to carry the sensor since even the UAVs were able to send the data back using the MOTE network. Other problems with UAVs was that they create significant pressure difference around the sensor making its data have a lot of noise. This was removed using a simple sliding average of the last five measurements.

Few extensions such as expanding measured data with dissipation to its surroundings using Gaussian dissipation nor fusing input from UAVs to main robots was not made due to time constraints and that the operator wanted as least data at the screen as possible to ease his cognitive load.

5 Datasets

During this work, three rounds of datasets were collected at differed times during the project. Each o those datasets was collected in a different environment with a different goal in mind to cover as much as possible scenarios ranging from perfect drives to real use case during SubT competition.

All of those datasets used previously mentioned JetsonTX2 and Sensorin SDC30 to collect the Wi-Fi and CO_2 data, respectively. Robot localisation used a large part of localisation pipeline of CTU-CRAS-NORLAB teams for DARPA SubT challenge which utilises robots odometry fused with onboard IMU. Additionally, a robot uses 3D lidar with iterative closest point (ICP) algorithm to correct drift in the odometry estimation. All of those methods together perform SLAM and can map the environment and localise the robot with a precision that did not, in most cases, influence the outcomes of the object localisation algorithm [43].

Alongside those three bigger datasets that were mainly used for Wi-Fi/CO_2detection and localisation, one small dataset was collected. This collection was done separately and included only sound data from robots onboard microphone with a phone located 1 to 5 meters from the robot. Multiple different pitched sounds were played at full volume from the speakers of the phone to simulate maximum capable sound created by the phone or survivor artefact.

5.1 First tests

First datasets were collected at CTU premises at Karlovo Namesti in Prague. Robot drove a small circuit on an outdoor area between the buildings and in the surrounding buildings. This data collection utilised Husky-A200 robotic platform which drove a circle with an LG-G6/H870 phone placed in the middle on a wooden bench. Ground truth data were collected by measuring phones position against a robots map by tape measure before robots start. Three separate setups were recorded and tested each representing a different real-world scenario. First path Fig:17 was encircling the bench at about 12 meters distance and then approached the phone to within 5 meters. This setup was done to simulate the best possible conditions for the robot to find the phone. This situation includes having the phone seen from all sides as well as outside where the n constant is the most stable since less multipathing occurs in this situation.

The next three datasets were done inside the campus building in hallways. Meaning of which was to simulate heavier obstructions and possible miss measurement due to blockages and signal interference since the building is filled with 2.4GHz transmitting devices. All those three paths were done inside a brick building with the same robot in a 2.7m wide hallways with doors, tables and other obstruction.

In the second dataset Fig:18 robot drove around the phone placed on the ground and around a corner near the phone simulating a general setup where the robot drives around the phone in the building. The goal was to evaluate if eventually, the phone is near the robot on a diverse path it can be detected and localised.

Third setup Fig:19 utilised driving around a room with the phone inside with a corner near the door of the room. Door to the room was closed to simulate a scenario where the phone cannot be detected by any line of sight (LOS) means to show that this localisation can add and improve LOS solutions. Another usage for this setup was to evaluate if it is possible to localise the phone in a different room altogether.

Last setup Fig:20 had phone placed on the ground on one side of the hallway near the metal elevator. Robot drove along the hallway without any significant turns. This setup represented a setting that would arise if the phone measurements were taken in an only straight line which is a setting any lateration has problems.

CO_2 sensory data were also collected in all three experiments, but no detectable higher levels of the gas were detected since all room, hallways (and outside) were ventilated enough.

Those experiments were used as basic testing and debugging dataset with an assessment of several methods used down the line in development.

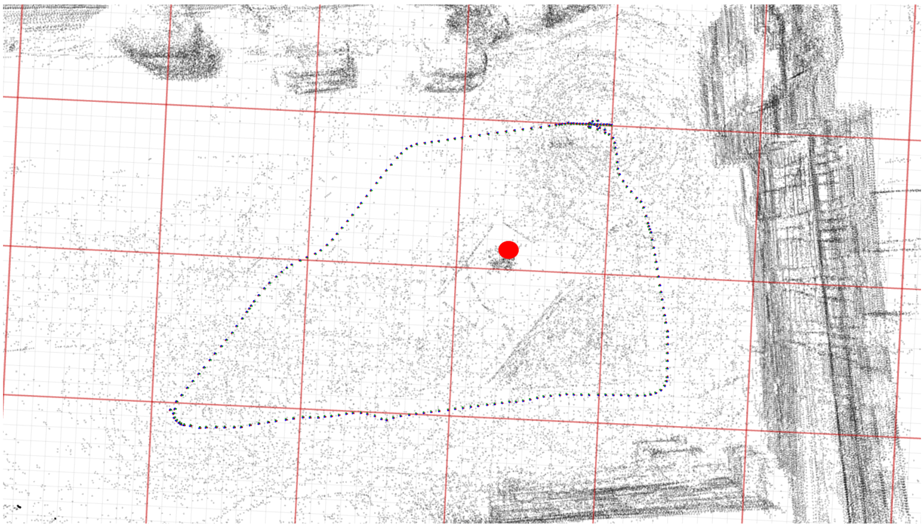


Figure 17: Robot outside map and path, with phone in middle

5.2 Prague underground

The second batch of datasets was firstly gathered for later processing. All of them were done in undisclosed underground infrastructure in Prague. This efferent consisted of concrete-metal reinforced structure that spanned several stories of metal equipment, making it hard for a signal to pass through. Two separate datasets were made where robot drove between multiple stories, entered small and big rooms and traversed stairs. Some

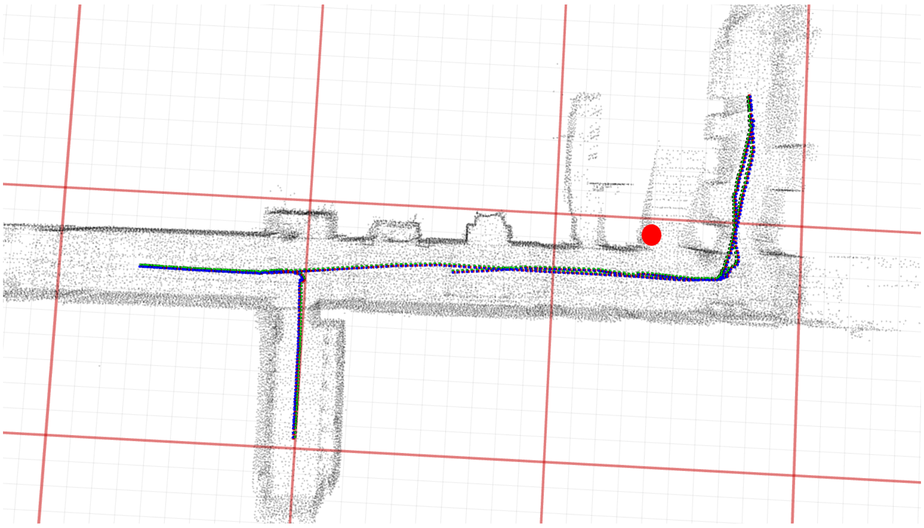


Figure 18: Robot map and path around a corner, with phone on the right corner

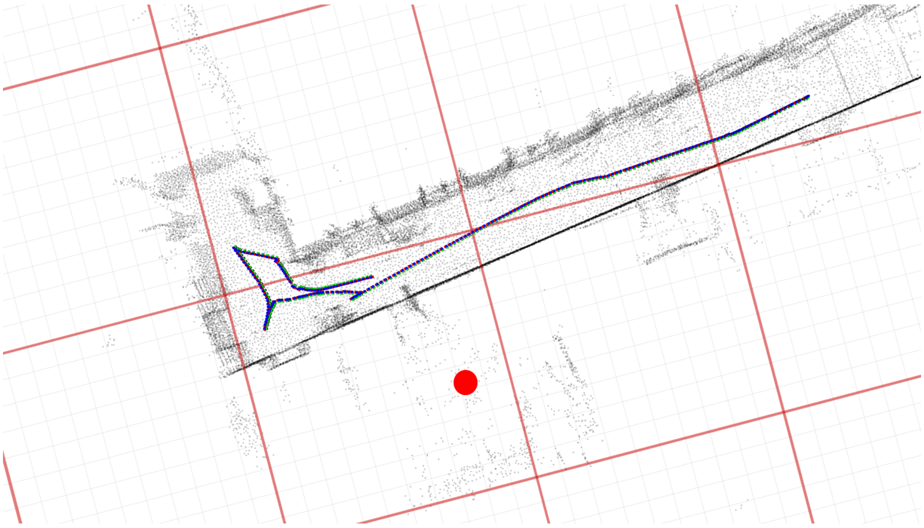


Figure 19: Robots map and path around a adjacent, closed room with a phone

special features included grated metal floor, multiple passageways between floors, large rooms with up to 10m ceilings and heavy blast doors.

During this dataset, most phones were placed by hand. Therefore, precision could not be measured since no ground truth about phone position was recorded. For any evaluation estimates of the phone, a position was acquired from other gathered data (robot localisation, camera, lidar) by hand.

Two Absolem robots were used in data gathering since the Husky cannot traverse stairs. The only difference from previous tests was that up to 3 different phones (LG-G6, iPhone SE, Samsung J6) were used at the same time.

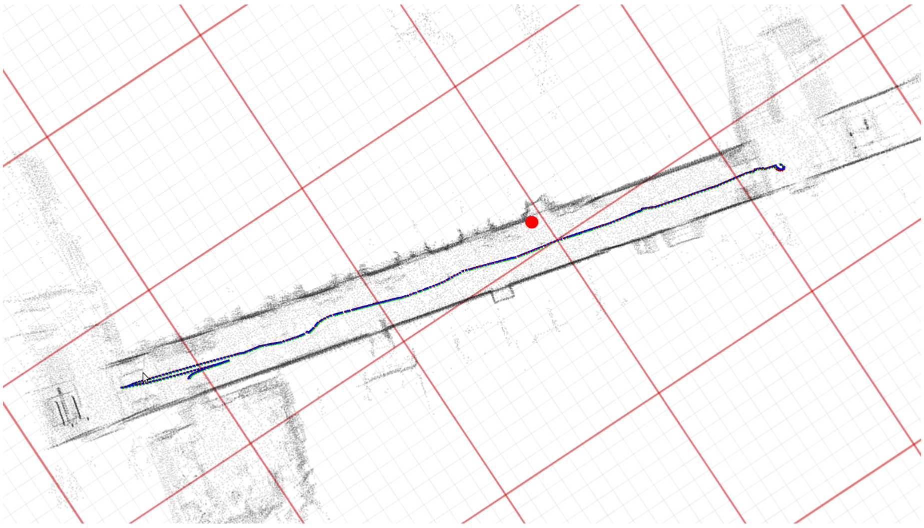


Figure 20: robots map and path along the hallway with a phone

More CO_2 measurements were taken during both runs, but the first run had corrupted data so it could not be used for any CO_2 analysis. To raise CO_2 levels to simulate the detections breathing directly onto the sensor was utilised due to a policy that forbids pressurised containers on the premises. This spiked the measurements up to 5000 ppm for a few seconds, which was well above the required 2000ppm.

5.3 Final deployment

During DARPAs SubT the Urban circuit took place in an unfinished nuclear power plant at Satsop Business park in Washington state, where one of the reactor buildings was divided for both tracks [53]. This environment consisted of thick metal-reinforced concrete walls, multiple stories with 10m between floors, water on the ground (up to 10cm), small passageways, large rooms with a ceiling of 15m and up, placed rubble, artificial smoke and haze, dust and frequently changing light conditions. Each team had a total of 4 runs, two on each track in 4 different days. Thus this dataset consists of four separate runs.

Day two and four were driven on a “Alpha” course and days one a three on “Beta” course. Courses did not change during each run, but artefact positions could have been changed.

On “Beta” the first 60 meters of the course was a long straight hallway with small rooms (3x2m) on sides, then robots could drive to the left where they had to choose if they want to go in the the outer perimeter of the building, climb stairs to the mezzanine above or go around the power plants reactor body.

On the first run on “Beta” all of our robots made it to the end of the hallway and then one went near the reactor. Other robot made it around the outer rector perimeter. All of the robots eventually got stuck on rubble, curbs or other debris in the vicinity. Second, run on "Beta" had similar outcome where the robots made it as far as last time except the one that went around the reactor made it about halfway around more than on the first run. Also during the second run, one of the robots tried to traverse the stairs up to the mezzanine where he managed to flip himself on his back due to hardware and traction failure on the stairs. All hardware survived except for few 3D prints and one mounting rod which was promptly replaced for a new one.

"Alpha" course had stair access right by the entrance gate which led down for 10m to the second floor where two hallways spanned for about 30 and 70 meters in two directions with lots of small rooms on sides. Robots also could go directly towards the reactor which was in the other direction that the stairs just right outside the starting gate. Around the reactor was a somewhat clean floor with large, almost cavernous, rooms with 15m ceiling with just few metal beams. Several larger and smaller rooms were adjacent to this large room with a entrance to power plant core itself.

During the first run on "Alpha", two robots went down the beginning stairs to search the lower floor where they were able to drive about 70m each. The other two robots were driven on the upper floor where one managed to get into the middle part of the reactor building, and the other has driven about half a circle around it. On the second run on "Alpha", only one robot went downstairs where he had the same coverage as previous two. another two robots covered the upper floor where they have driven about 50% more than in first, run both around the core structure. The last large robot had problems with alignment to the map in which the artefacts had to be reported, so he was eventually just used a relay station for the others without any gathered data.

During all of those runs, two more hexapods and drones were deployed with various levels of sauces, but none of those had any detections sent back to the base. In later runs some of those smaller robots were converted to mobile retransmission stations for Mobilicom and MOTE networks.

During the first three runs problems with collecting Wi-Fi data arose from Jetson TX2, which was recording all with the wrong timestamp. This was due to the system starting without any internet connection and the Jetson not having a real-time clock with battery. This problem was attempted to be fixed for the third day, but the bug-fix showed to be still not working. This error led to the Wi-Fi data having timestamp offset of two years for the first two runs and 45 minutes for the third run. Those problems were later fixed by manually aligning the data. Another problem was caused by different software on robot Husky which did not allow for recording the collected Wi-Fi data resulting in none of it from this very robot altogether.

Altogether the robots covered about 2.5 Km of tracks during four runs.

Ground truth to all the measurements was later provided by DARPA staff [54]. It consisted of a dense coloured point-cloud scan of the whole building as well as location and type of each artefact with precision to 1mm.

Table 5: KPI comparison on first dataset with object map and lateration approaches

Dataset	Method	E [m]	$\overline{E}$ [m]	$RMSE$ [m]	C [-]
Outside	Object map	2.20	8.00	9.01	31
Corner	Object map	0.87	1.12	1.27	65
Room	Object map	1.14	6.69	11.10	62
Hallway	Object map	1.06	12.78	17.27	97
Outside	Lateration	2.63	8.67	11.13	29
Corner	Lateration	0.87	6.46	12.20	11
Room	Lateration	0.53	3.19	4.55	11
Hallway	Lateration	25.40	52.3	57.7	-

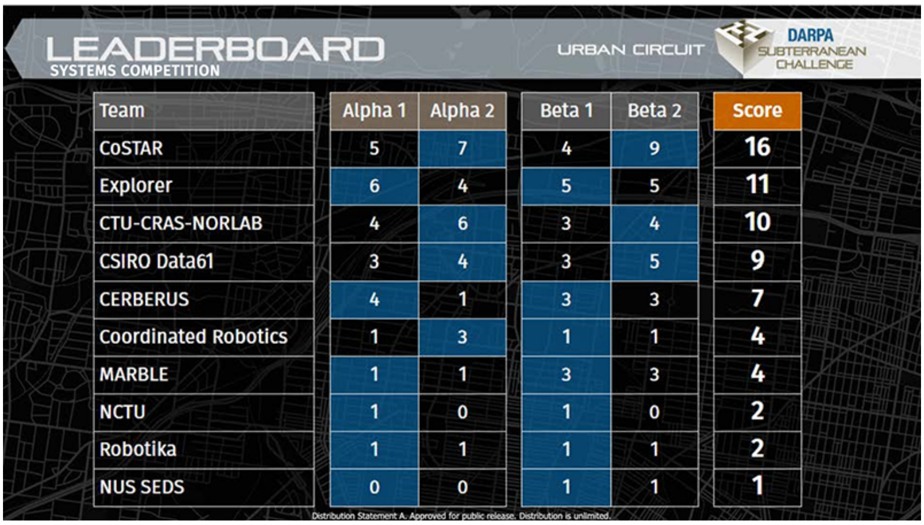
6 Results

All datasets except the one from Prague underground which has to remain secret were evaluated against the KPI that is described at 3.3. Two evaluations were done where constants n and a from equation7 were firstly estimated with minimal prior knowledge of the environment and then tuned for each scenario to show the best achievable values.

The constants were tuned by hand and could still be probably tuned a bit better. During the first four experiments, only one phone was present and was always detected however in the final competition not all phones were. To “detect” an artefact a robot had to know about its existence and its BSSID. The localization was not performed on any data that had a signal strength of -80 dBm or worse since the variance of the distance was too high. Final estimation of phone positions was limited to 4 guesses total since the maximum amount team can send to the DARPA team is also limited to 40 for a total of 20 artefacts. CO_2 detections had six tries and are done with accordance to the set DARPA rules requiring 2000 ppm on the spot the robot is located at. To accommodate for not properly calibrated sensors, the threshold value was set to 1800 ppm during the run.

Sound detection proved to be impossible since the phone was barley hearable in robots microphone. This was mostly due to noise the robot makes by its cooling fans and other moving parts even while it is standing.

In the end, choices to implement properly only one approach to the whole pipeline were driven by limited computational resources and the request of the system operator. The request was to only see one estimate at the time making the only advantage of object map irrelevant. The final implantation helped our team score 2 points which would otherwise resulted in worse placements Fig:21. On the other hand with just a little better localization we would be able to acquire two more pints shifting us to second place.



Team	Alpha 1	Alpha 2	Beta 1	Beta 2	Score
CoSTAR	5	7	4	9	16
Explorer	6	4	5	5	11
CTU-CRAS-NORLAB	4	6	3	4	10
CSIRO Data61	3	4	3	5	9
CERBERUS	4	1	3	3	7
Coordinated Robotics	1	3	1	1	4
MARBLE	1	1	3	3	4
NCTU	1	0	1	0	2
Robotika	1	1	1	1	2
NUS SEDS	0	0	1	1	1

Figure 21: Final scores and placements of the SubT Challange Urban round with CTU-CRAS-NORLAB placing third, first among self funded team

6.1 First tests

Comparing results Table:5 from the first dataset neither approach has a significant edge over the other. Object map approach was able to find the object in all cases but took significantly more data to do so. The only setting that proved to be difficult for multilateration is the hallway where the robot was not able to estimate any position of the object precisely. Results from this test showed that both approaches could be used for alter computation in and that the only dilemma was to choose in between required data and a problem with straight hallways. In all other tests, both systems performed virtually the same and the performance depends on the trajectory the robot makes during the environment scouting.

6.2 Prague underground

Even though data gathered in those experiments were not included at performance analysis since the data cannot be released for the public to validate they have been beneficial to get other insights. Since the data are from a real multistory structure with large ground footprint showed a need for true 3D localization which was not done by the object map approach. It also showed that multiple maps need to be created at the same time, one per phone for which the scalability of this solution did not suffice. It was also shown that it needed about 800MB of RAM per 200×200 floor per phone. This fact eventually resulted in polishing only the multilateration application while being accepted that the localization will not work correctly if the robot drives only in a straight line which was considered improbable during the final deployment. Another insight we have acquired was that in

Table 6: KPI comparison on the final deployment dataset

Dataset	E [m]	$\overline{E}$ [m]	$RMSE$ [m]	C [-]	P_{Wi-Fi} [% -]	P_{CO_2} [% -]
Run "Beta 1"	4.89	20.83	29.70	21	25% 4	33% 1
Run "Alpha 1"	20.66	49.32	50.67	-	100% 4	66% 3
Run "Beta 2"	11.20	16.81	17.80	-	25% 4	33% 4
Run "Alpha 2" CTU	8.21	12.70	12.80	-	100% 4	66% 3
Run "Alpha 2" TNO	16.09	27.85	28.69	-	100% 4	66% 3

environments like those structures, the n value varies drastically depending given layout and even through floor types.

6.3 Final deployment

During the competition, the system was properly running only on the last day but all data were later analyzed to evaluate the pipeline against all the KPIs. This evaluation was done using DARPA ground truth as well as a handy map which shows positions of all the artifacts. This map is included in the appendix 9 of this work and all later references to specific artefacts are labelled according to this map. Evethought the results in the Table:6 look much worse then in previous experiemts our team was able to score 3 points two of which were counted agianst our final score and helped us to be in first three teams.

6.3.1 Beta

From Table:6 can be seen that beta course was harder for our system to even detect any phones altogether but due to the placement of the first phone being directly near the entry hallway the ones that were detected were localised with good precision. This is true especially for the "Beta 1" where the detected phone (62C-) was in a similar setup as in the "Corner" dataset but only on top of mezzanine making the localization quite precise. On the other hand during "Beta 2" the phone (92-C) was placed on top of a barrel along the main hallway which effectively made it as in scenario "Hallway". This only confirms that the multilateration has quite problems with measurements that are taken exclusively on one line.

6.3.2 Alpha

On the alpha course, each of the runs had all phones detected at least once by at last one robot which was somewhat easier to do since the lower floor was much more easily accessible than the lower floor of beta course which hid two phones per run. Even though all of them were detected during “Alpha 1” most of them (34C-, 8C-) was seen only by one robot that again went only in a straight line and when he finally was ordered by the operator to turn his the localization was lost, which meant no more usable data. Additionally, only one phone (43CB) was on the lower floor during this round where two of the robots went, and its location was on the far side of the building in which direction none of the robots went. Contrary to this debacle “Alpha 2” run had one phone (40-C) really close to the stairway access to the lower floor. This artefact was detected as well as all other in that run and was localized with an error under 10 meters which was less than 5m short to be enough for scoring. The robot went several times around the location of this phone, but since the phone was embed ed in a small recess in a wall just below the 8m ceiling, it would require an elevated vantage point to make a LOS contact with it. The most probable reason why the localization failed, in this case, was a reflection problem which caused signal from this point to be stronger a little more down the hallway on a ground level than it actually was.

The other phone was detected (15-C) and localized by the sole robot that went around the core of the building. His path was not strictly straight but did not include large turns either. The phone was localized with a precision of 15 meters with an error made in the general direction from the robot at that time, making it at least somewhat useful as a direction one can try to explore. The robot would have to drive at least another half of what it had driven altogether to get to its position since it was located behind a corner of a somewhat long hallway which itself was behind 2m thick wall. In the end robot had it about 5m away from the phone but had the wall in between.

Altogether results in TABLE show that Both object map and Lateration are able to localize some of the artefacts, but it highly depends on the scenario. Object map generally performs a little better compared to the Lateration and can find the object in the hallway scenario, albeit needing a large amount of data.

6.3.3 CO_2

CO_2 detections were evaluated mostly during the final deployment since the previous experiments had an only small amount of data corresponding to the artefact. During the run, the detections were sent back to base with the MOTE network, but the operator could still see all measured data and set the threshold manually.

Since not all possible guesses by the CO_2 detections were used, the operator could have used those to correct other detections which have paid off since we have run out of possible guesses on the run “Beta 2” where the operator managed to score point about 5 seconds before the end with the last available guess.

During the “Beta 1” only one gas artefact (57GG) was detected properly and reported by the system. Another gas (64G-) was in a small room next to the entrance hallway that was not detected. This was due to the robot nod diverting from the main hallway and because the actual concentration in the hallway was not much higher than the ambient concentration in other places (700 ppm). The same case was in the “Beta 2” run where the first artefact (57GG) was detected again with a different robot and similarly placed artefact (67-G) in the small room next to the hallway was not detected again. During this run, four detection attempts were made from which three were done in one place where one of the robots flipped on its back. All three incorrect measurements were done by this robot, and one possible explanation is that the senor was closer to the ground, which is colder and wetter affecting the measurements.

With “Alpha 1” run two detections were made but with an error from ground truth of 5.6 (32G-) and 6.2 (26G-)meters from their respective artefacts. This made the reports to the DARPA team not precise enough. Most probable cause is that the gas was dispersing too slowly in this environment and caused to be detected too far from the actual room entrance. One common feature of those two artefacts was that both had an entrance from a large open room. Another detection attempt was made by one of the robots that went downstairs where it detected elevated CO_2 gas at the position of artefact (41-G) which strangely was not meant for this round on this track. The most probable cause is that the infrastructure for the artefact itself was in place releasing a smaller amount of gas which was under 2000 ppm but above the aforementioned threshold of 1800ppm. This might have been done to sway the searching robot from the actual gas artefact (42G-) that was located several more rooms down the hallway.

On “Alpha 2” track one artefact (35-G) was detected and properly reported to DARPA. Another artefact (19-G) was detected correctly, but the position reported to DARPA was not precise enough since it had an error from ground truth 5.2 m. The cause could be the same as in the two previously mentioned artefacts which were also with an ingress point from this large room. The only robot that went down the stairs did not detect the gas (41-G) that was previously there since it did not go into the small room again. Another wrong detection was made two floors above artefact (74GG) which was actually on a different track with each floor having a hatch near this place. This was most likely just a coincidence since the CO_2 gas should not rise upward.

6.4 Gained insights

To improve Wi-Fi localization capabilities implementations and experiments with the better acquisition of constant n would help immensely. One possibility is incorporating values from between the robots whose positions are known. This would allow us to compute the n value for each specific room.

More uniform detection range would help the multilateration.

Usage of object maps might also prove to be working better than the multilateration approach if the memory size problems are solved with some kind of ingenious architecture. This is due to the fact that they do not suffer from problems with long straight pathways. Another possible addition would be to redo the whole probability map approach using only some kind of limited map size movable trough space.

A different solution would be to use a hybrid system where the multilateration is default but if the robot goes for too long in on one line system might try to calculate reduced-size map with object map approach.

For CO_2 detection, better calibration of CO_2 sensors t hat would have the exact values on each robot would help to make the estimates better. Improving the CO_2 detection algorithm with the usage of gradient or some other kind of algorithm that does better analysis then just threshold would help with places where the CO_2 level is just slightly elevated about its ambient vales at that place. The most straightforward upgrade to the CO_2 algorithm would be to measure all values at one place and then take the strongest one in the bunch, not the first that is done now. This would eliminate the mistakes with detecting the gas too far away from the actual artefact.

7 Conclusion

The main goal of the work was to implement a system with the use of multiple sensors to find and locate objects in the adversarial environment and to implement the solution with DARPA SubT Challenge in mind.

The system is comprised of two parts where the smaller is for detection and localization of elevated CO2 levels and the other for the detection of mobile phones.

The Wi-Fi signal strength (RSSI) is used to solve the phone detection problem in conjunction with a robot's position to detect devices in adverse conditions where the communications, light, or any other infrastructure might not be present. Implementation of this solution at the beginning of this work with object position map where the measurements were all incrementally incorporated into a larger map of the whole world which showed to be not fast enough and using too many resources to be considered for final deployment. As a solution to this problem choice was made in favour of multilateration approach. This approach solves the problem analytically making it light on CPU and RAM usage compared to the previous method. With the usage of weighting and chunking, the method was implemented to be able to calculate the 3D position from just a list of measurements and their positions to allow for smaller data bandwidth needed.

This system was implemented using ROS with a combination of C++ and Python on robots that are part of CTU-CRAS-NORLAB team in SubT Challenge competition.

Testing and validation was separated into three parts where first tests served as proof of concept, second as a test in a more realistic environment in Prague underground, and last was real deployment in second "Urban" round of DARPA Subterranean Challenge which took place in an old nuclear power plant in Washington state, US.

Continuous testing showed main weakens of the multilateration method, which has problems with the solution when the measurements are taken on one line.

In the last deployment, the CO_2 detection proved to be valuable and allowed our team to score multiple points which would be otherwise unavailable to us.

The phone detection was working only on later days of the competition in which the system was able to detect the phone, but the localization was not precise enough to improve our scoring even-though with recomputed data from the first run it would help us get one more point. This presents new improvement possibilities into the next rounds in this challenge where the system will be deployed again.

Implementation of this system allowed our team to acquire more points without which we would not be able to stand on the winning podium. The third place overall and first among self-funded teams brought us a prize of 500K USD.

8 Future work

In the future, possible addition of a thermal camera to the system would be a welcomed addition allowing for detecting several artefacts in non-lit scenarios.

Moreover, implementation of the phone measurements to the visual detection as a prior might help with a problem of false positives in that solution. This could be even done the other way where the propagation constants are derived from data that visual detection makes.

Addition of omnidirectional antenna that has the radiation pattern as a sphere would allow for the inclusion of directional data compared to an only range that is used now.

References

[1] DARPA. Competition rules urban circuit, November 2019.

[2] Geert De Cubber, Daniela Doroftei, Daniel Serrano, Keshav Chintamani, Rui Sabino, and Stephane Ourevitch. The EU-ICARUS project: Developing assistive robotic tools for search and rescue operations. In *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, October 2013.

[3] Ivana Kruijff-Korbayová et al. TRADR project: Long-term human-robot teaming for robot assisted disaster response. *KI - Künstliche Intelligenz*, 29(2):193–201, February 2015.

[4] L. Marconi et al. The SHERPA project: Smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments. In *2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, November 2012.

[5] M. Bijelic et al. A benchmark for lidar sensors in fog: Is detection breaking down? In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 760–767, 2018.

[6] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, March 2018.

[7] S. Yang and C. Wang. On solving mirror reflection in lidar sensing. *IEEE/ASME Transactions on Mechatronics*, 16(2):255–265, 2011.

[8] F. Majer et al. Learning to see through haze: Radar-based human detection for adverse weather conditions. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–7, 2019.

[9] A. Pfeuffer and K. Dietmayer. Optimal sensor data fusion architecture for object detection in adverse weather conditions. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–8, 2018.

[10] S. G. Tzafestas. *Introduction to mobile robot control*. Elsevier, London Waltham, MA, 2014.

[11] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[12] GHO — by category — exposure limits for radio-frequency fields (public) - data by country.

[13] R. Hickling and A. W. Brown. Determining the direction to a sound source in air using vector sound-intensity probes. *The Journal of the Acoustical Society of America*, 129(1):219–224, January 2011.

[14] Raúl Guzmán-Quirós et al. Integration of directional antennas in an RSS fingerprinting-based indoor localization system. *Sensors*, 16(1):4, December 2015.

[15] Hans-J von der Hardt et al. A method of mobile robot localisation by fusion of odometric and magnetometric data. *The International Journal of Advanced Manufacturing Technology*, 9(1):65–69, 1994.

[16] Lindsay Kleeman et al. Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning. In *ICRA*, pages 2582–2587, 1992.

[17] Adrian Ratter et al. GPU accelerated graph SLAM and occupancy voxel based ICP for encoder-free mobile robots. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 540–547. IEEE, 2013.

[18] Radu Bogdan Rusu et al. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[19] Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[20] Armin Hornung et al. Octomap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[21] David J Austin and Brenan J McCarragher. Geometric constraint identification and mapping for mobile robots. *Robotics and Autonomous Systems*, 35(2):59–76, 2001.

[22] David Filliat and Jean-Arcady Meyer. Map-based navigation in mobile robots:: I. a review of localization strategies. *Cognitive Systems Research*, 4(4):243–282, 2003.

[23] Andrzej Pronobis. *Semantic mapping with mobile robots.* PhD thesis, KTH Royal Institute of Technology, 2011.

[24] Tim Bailey et al. Consistency of the ekf-slam algorithm. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3562–3568. IEEE, 2006.

[25] Robert Sim et al. Vision-based SLAM using the rao-blackwellised particle filter. In *IJCAI Workshop on Reasoning with Uncertainty in Robotics*, volume 14, pages 9–16, 2005.

[26] Masahiro Tomono. Robust 3D SLAM with a stereo camera based on an edge-point ICP algorithm. In *2009 IEEE International Conference on Robotics and Automation*, pages 4306–4311. IEEE, 2009.

[27] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, October 2015.

[28] Wolfgang Hess et al. Real-time loop closure in 2D LIDAR SLAM. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278. IEEE, 2016.

[29] Tomáš Krajník et al. Fremen: Frequency map enhancement for long-term mobile robot autonomy in changing environments. *IEEE Transactions on Robotics*, 33(4):964–977, 2017.

[30] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[31] Michael Buckland and Fredric Gey. The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12–19, 1994.

[32] Tom Duckett and Ulrich Nehmzow. Mobile robot self-localisation and measurement of performance in middle-scale environments. *Robotics and Autonomous Systems*, 24(1-2):57–69, 1998.

[33] Wei-Yu Chiu et al. Minimum manhattan distance approach to multiple criteria decision making in multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 20(6):972–985, 2016.

[34] Margrit Betke and Leonid Gurvits. Mobile robot localization using landmarks. *IEEE transactions on robotics and automation*, 13(2):251–263, 1997.

[35] Lars Krüger and Christian Wöhler. Accurate chequerboard corner localisation for camera calibration. *Pattern Recognition Letters*, 32(10):1428–1435, 2011.

[36] Jan T Bjørke. Framework for entropy-based map evaluation. *Cartography and Geographic Information Systems*, 23(2):78–95, 1996.

[37] University-Corpus Christi. Disaster city range, May 2020.

[38] Sören Schwertfeger. *Robotic Mapping in the Real World: Performance Evaluation and System Integration*. PhD thesis, 07 2012.

[39] T. Krajník et al. External localization system for mobile robotics. In *16th International Conference on Advanced Robotics (ICAR)*, Nov 2013.

[40] Jürgen Sturm et al. A benchmark for the evaluation of RGB-D SLAM systems. pages 573–580, 10 2012.

[41] DARPA. Artifacts specification cave circuit, April 2020.

[42] Saska et al. DARPA subt STIX qualification submission: CTU-CRAS.

[43] Tomáš Rouček et al. Darpa subterranean challenge: Multi-robotic exploration of underground environments. In *International Conference on Modelling and Simulation for Autonomous Systesm*, pages 274–290. Springer, 2019.

[44] Manikanta Kotaru et al. Spotfi: Decimeter level localization using wifi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 269–282, 2015.

[45] George Broughton et al. Rfid-based object localisation with a mobile robot to assist the elderly with mild cognitive impairments. In *Intelligent Environments (Workshops)*, pages 366–375, 2016.

[46] Luis Oliveira et al. RSSI-based relative localisation for mobile robots. *Ad Hoc Networks*, 13:321–335, 2014.

[47] Abdalkarim Awad et al. Adaptive distance estimation and localization in wsn using rssi measures. In *10th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD 2007)*, pages 471–478. IEEE, 2007.

[48] Tamer Dag and Taner Arsan. Received signal strength based least squares lateration algorithm for indoor localization. *Computers & Electrical Engineering*, 66:114–126, 2018.

[49] Theodore S Rappaport et al. Propagation and radio system design issues in mobile radio systems for the glomo project. *Virginia Polytechnic Institute and State University*, 1997.

[50] Wolfram Burgard, Dieter Fox, et al. Estimating the absolute position of a mobile robot using position probability grids. In *Proceedings of the national conference on artificial intelligence*, pages 896–901, 1996.

[51] Jeong Heon Lee and R Michael Buehrer. Fundamentals of received signal strength-based position location. *Handbook of position location: Theory, practice, and advances*, pages 359–394, 2011.

[52] Abdelmoumen Norrdine. An algebraic solution to the multilateration problem. In *Proceedings of the 15th international conference on indoor positioning and indoor navigation, Sydney, Australia*, volume 1315, 2012.

[53] Unearthing the subterranean environment. `https://www.subtchallenge.com/`. [Online; accessed 21-May-2020].

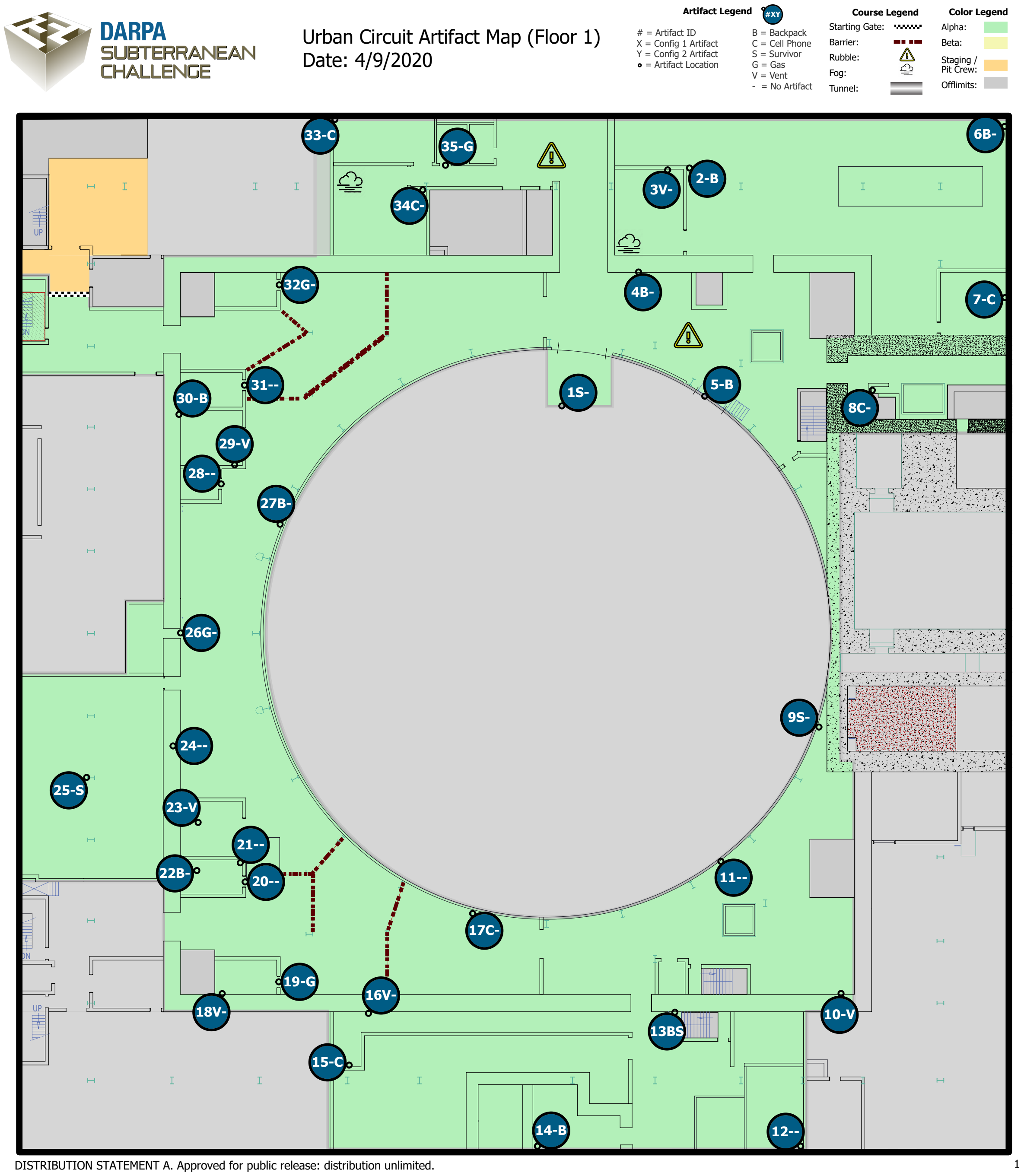
[54] DARPA SubT urban ground truth. `https://bitbucket.org/subtchallenge/urban_ground_truth/src/master/`. [Online; accessed 21-May-2020].

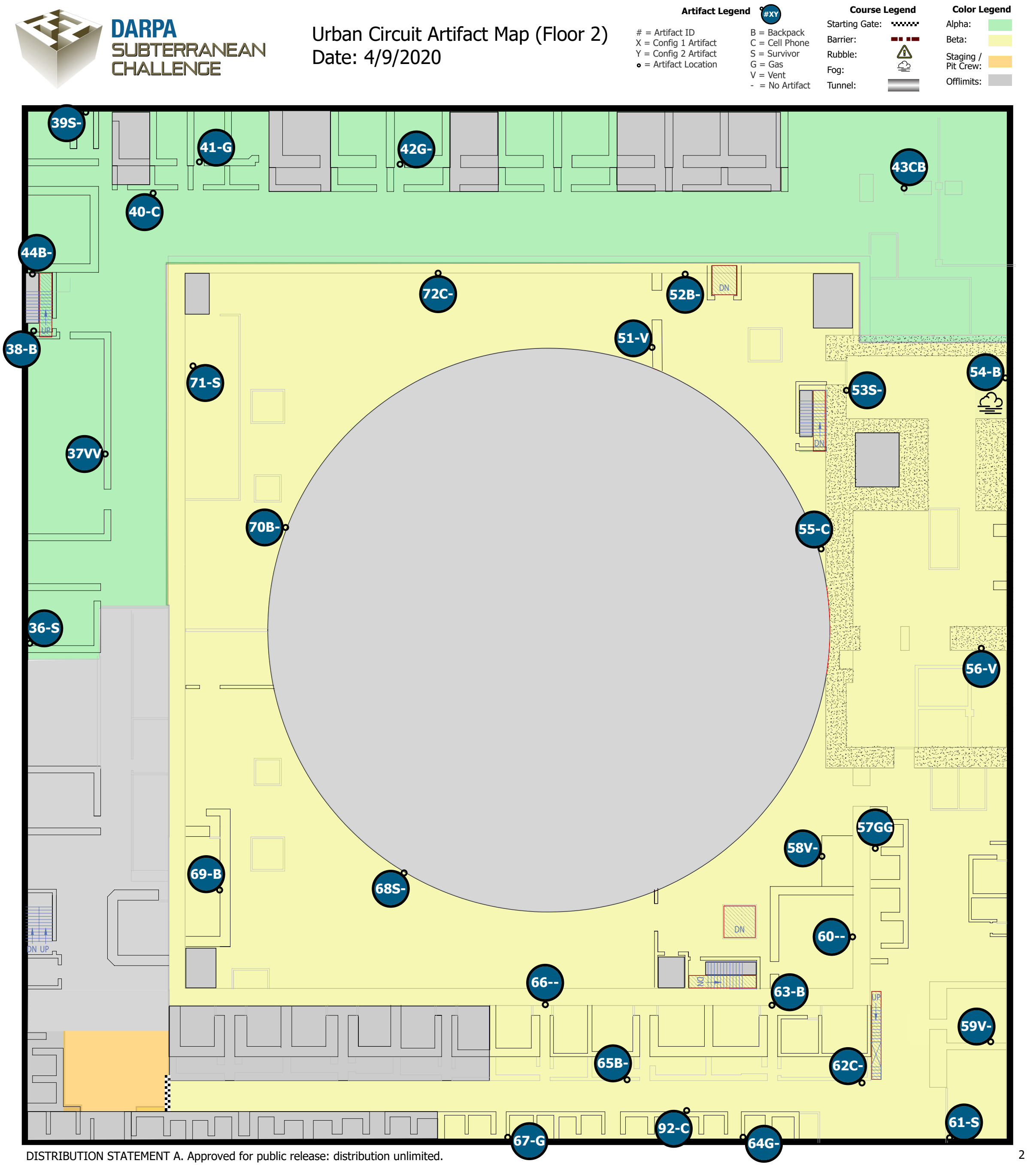
9 Appendix

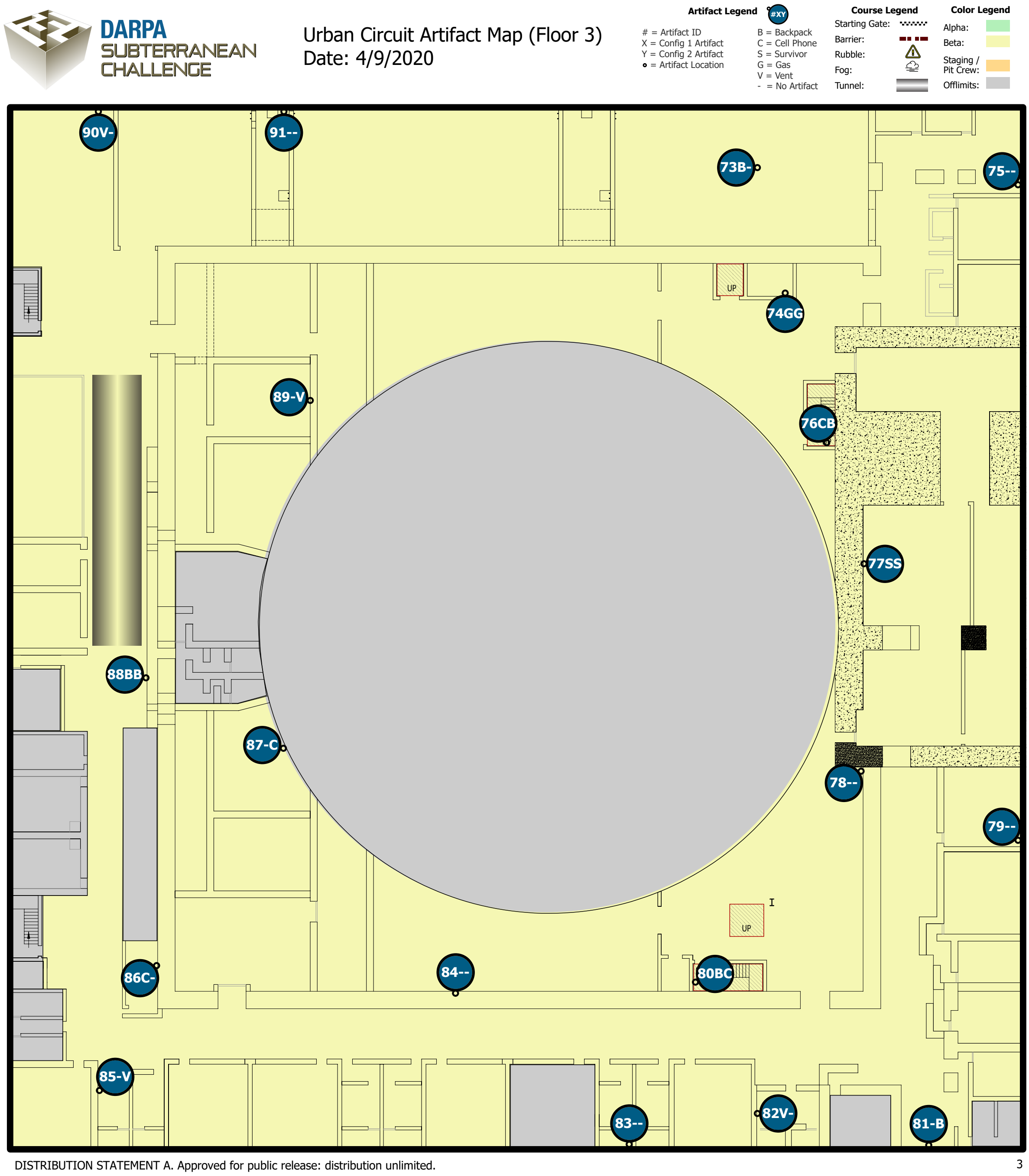
Directory name	**Description**
text	this thesis in pdf format
sources	source codes
datasets	all releasable datasets

Table 7: Attached CD contents

Site and artifact ground truth




DARPA
SUBTERRANEAN
CHALLENGE
Urban Circuit Artifact Map (Floor 2)
Date: 4/9/2020
Artifact Legend
#XY
# = Artifact ID
X = Config 1 Artifact
Y = Config 2 Artifact
= Artifact Location
B = Backpack
C = Cell Phone
S = Survivor
G = Gas
V = Vent
- = No Artifact
Course Legend
Starting Gate:
Barrier:
Rubble:
Fog:
Tunnel:
Color Legend
Alpha:
Beta:
Staging / Pit Crew:
Offlimits:
39S-
41-G
42G-
43CB
40-C
44B-
72C-
52B-
38-B
51-V
71-S
54-B
53S-
37VV
70B-
55-C
36-S
56-V
57GG
58V-
69-B
68S-
60--
66--
63-B
59V-
65B-
62C-
61-S
67-G
92-C
64G-
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.
2


DARPA
SUBTERRANEAN
CHALLENGE
Urban Circuit Artifact Map (Floor 3)
Date: 4/9/2020
Artifact Legend
#XY
# = Artifact ID
X = Config 1 Artifact
Y = Config 2 Artifact
= Artifact Location
B = Backpack
C = Cell Phone
S = Survivor
G = Gas
V = Vent
- = No Artifact
Course Legend
Starting Gate:
Barrier:
Rubble:
Fog:
Tunnel:
Color Legend
Alpha:
Beta:
Staging / Pit Crew:
Offlimits:
90V-
91--
73B-
75--
74GG
89-V
76CB
77SS
88BB
87-C
78--
79--
84--
80BC
86C-
85-V
83--
82V-
81-B
UP
UP
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.
3