CoCoMaps Project

CMLabs | IIIM

CoCoMaps Demo-1 Report

Demonstration date: 27 November 2017

Link to video on Youtube: https://youtu.be/maAh8N3nRFg

EXECUTIVE SUMMARY

This report describes the results of Demo-1 of the CoCoMaps Project, which is a joint effort by CMLabs (UK) and the Icelandic Institute for Intelligent Machines (Iceland). The aim of Demo-1 is listed as Milestone 4 in the agreed proposal and shows off the following deliverables:

- T8.D2 Draft Collaborative Cognitive Map
- T9.D1 Demo 1: Collaborative Visual Detection

The goal of Demo-1 is to demonstrate the advances made in the development of the Collaborative Cognitive Map architecture where two or more independent robots can work collaboratively by sharing and comparing live observations, negotiating the execution of tasks and finally make their own decisions on what to do next.

The CoCoMaps architecture is based on the Cognitive Map Architecture that was developed by CMLabs in collaboration with Honda Motors Research Institute (HRI) in California. The CoCoMaps project aims to extend the capabilities of the CMA, which worked for a single robot interacting with a single human, having the ultimate goal of having a new improved architecture that works for multiple robots and multiple humans, allowing them to communicate and collaborate. The project has four parts, the first being a demo of a simple virtual agent (Demo-0), the second involving two robots collaborating on visually searching for humans in the scene (Demo-1, this report), the third adding basic human-robot communicative capabilities allowing the robots to extract information from humans using natural dialogue (Demo-2, the next demo), and the fourth and final extending this by allowing robots and humans to talk about multiple pieces of information with dynamic feedback from the humans (Demo-3, to be done after Demo-2).

This report describes the successful conclusion of Demo-1, detailing data showing how collaborative visual search, human detection and recognition, and navigation are integrated in a running system involving two robots and humans present in the robots' area of operation. The results of processing times, CPU loads, and overall architecture reliability are within acceptable ranges, providing a foundation for continuing onto the next steps of the project and providing a valuable guide for the work to be carried out now.

INTRODUCTION

The overall goal of CoCoMaps is to demonstrate that our Cognitive Map Architecture (CMA) can be extended from single robot-human relatively simple interaction to multi-robot, multi-human more dynamic and social interaction. Getting to that final version of CoCoMaps in this project involves developing several sub-components which must be tested and demonstrated thoroughly to support continuing development. Demo-1 aims at demonstrating basic collaboration capabilities, integrated with navigation and appropriate visual competencies where two robots work in a people-sparse environment requiring detection of humans. The robots will collaborate on dynamically optimising the visual search for humans entering the scene. Specifically, the robots work together – collaborate and communicate – with a goal to reduce the amount of visual overlap, i.e. duplicated work, reducing the efficiency of the robots' tasks. The collaboration involves mutual communication about their observations and negotiating behaviours that are both time- and context-dependent. We test this by running scenarios that test key aspects of these capabilities. To ensure consistency and data reliability we use a partially-scripted scenario that is run several times in the same area. To evaluate the collaborative aspects each scenario is run with the robots in "solo" mode (each without any knowledge of what the other robot is doing) as well as with the robots working together ("collaboration" mode).

The goal of Demo-1 is to demonstrate the successful design and implementation of collaboration between the robots.

The KPIs (Key Performance Indicators) relevant for Demo-1 are used as guidelines (see Table 1 below).

To assess the performance achieved during Demo-1 a number of indicators are measured. These are summarised in Table 2 in the following section Experimental Setup.

The rest of the report is organised as follows: Following Experimental Setup we present the Results of Demo-1, which is based around numbers collected from multiple runs of identical scenarios intended to provide reliable evaluation of system performance on the listed KPIs, followed by Discussion & Future Work.

Table 1.
KPIs from CoCoMaps proposal.

2	Ability of real-world robot-robot interaction using new collaborative CMArch	M13	One Turtlebot able to see, listen and speak in simple setup	Two Turtlebots able to communicate via CMArch	Video recording, statistics graphs
4	Efficiency of collaborative detection of humans	M16	Initial measurement of detection efficiency at current SOA implementation	Measurement of detection efficiency at Demonstration 1	Measure added efficiency (speed, effort, error rate) of collaborative detection
5	Efficiency of collaborative tracking of humans	M16	Initial measurement of tracking efficiency at current SOA implementation	Measurement of tracking efficiency at Demonstration 1	Measure added efficiency (speed, effort, error rate) of collaborative tracking

EXPERIMENTAL SETUP

This section provides a description of, in the following order, physical space, robot hardware, robot software, measurements, and experimental procedure / run.

PHYSICAL SPACE

The demonstration took place in IIIM's offices in Reykjavik within an area of approximately 3 x 10 meters. The lighting was provided using built-in overhead fluorescent lights.

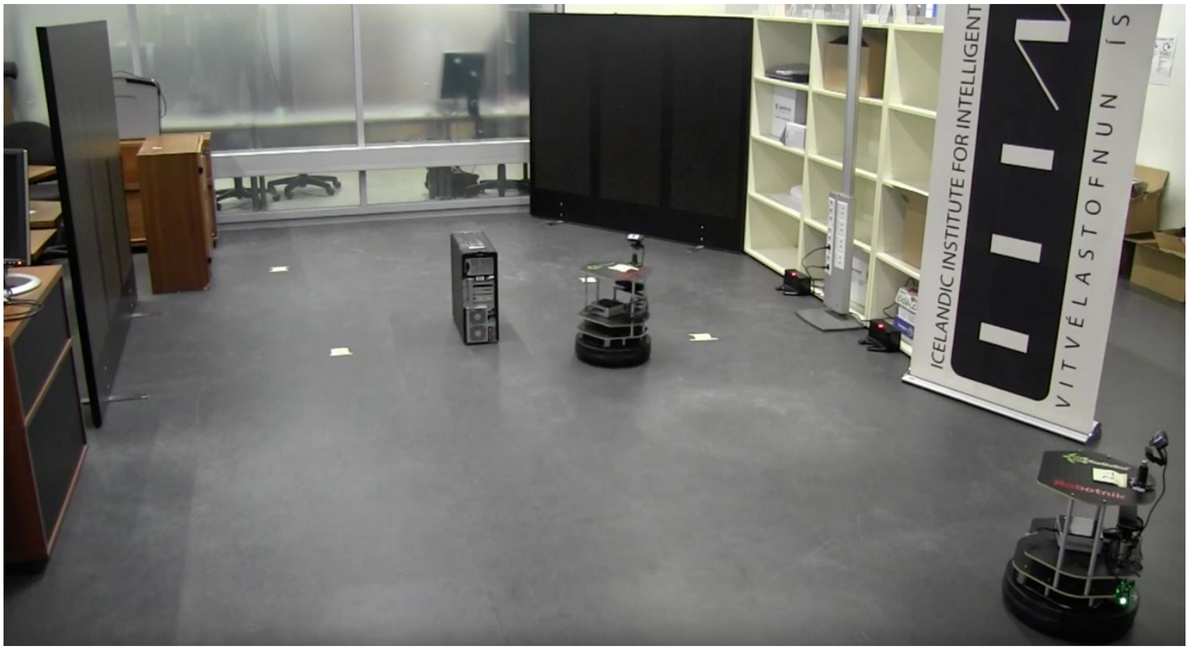

Figure 1. Experimental setup for CoCoMaps Demo-1. Here the two robots can be seen in the center and in the lower right-hand corner. The box to the left of the center robot serves as an obstacle to demonstrate the ability of the robots to perform their collaboration and human-robot interaction while avoiding obstacles and navigating the space. White markings on the floor indicate significant waypoints for the robots during their collaboration. The white markings are for human visual aide only and are not used directly by the robots in the demo. In the future these may be automatically generated at runtime.

DEMO-1 ROBOT HARDWARE

The robots used in this work are setup and arranged identically. Their hardware is identical in all aspects. The chosen setup is the TurtleBot 2 design, an open source hardware project that delivers most of the required components for fast setup and integration. The TurtleBot 2 is built on a Kobuki base, a mobile research base. The base supplies power for the entire system, has a motor to move through the surroundings as well as sensors used in navigation. TurtleBot 2 comes with setup for a 3D depth camera that can be used for mapping and localization. For the main control a computer is placed onto the TurtleBot structure with wifi capabilities to control remotely. For human recognition an additional USB camera is placed on top of the structure.

The complete structure is cylindrical with a diameter of 354 mm and height, from floor to top of the structure 420 mm. The Kobuki base has ground clearance of 15 mm. The combined weight of the base and structure is 6.3 kg, without the computer, USB camera and other additional peripherals.

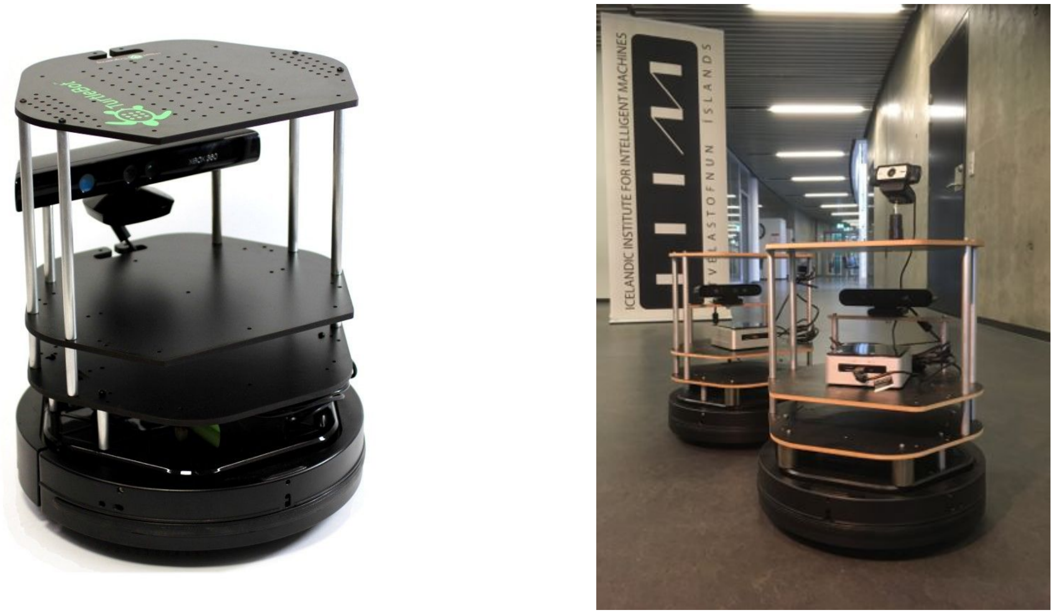

Figure 2.
Left: TurtleBot 2 structure assembled on the Kobuki base, including an Astra Orbbec 3D depth camera. *Right:* With control computer and the USB camera added.

The Kobuki base uses a standard 12 V brushed DC motor. The batteries are Lithium-Ion 14.8V 4400 mAh, 4S2P configuration. Additional sensors used in navigation are a 3-Axis digital gyroscope from STMicroelectonics, part name L3G4200D, with a measurement range ±250 deg/s. Additionally the base comes with 3 bumper sensors, left, center, right.[1]

For navigation, mapping and localizing a 3D depth camera, Astra Orbbec, is placed in the center platform of the TurtleBot structure. The camera has a range of 0.6-8.0 m with a maximum depth image size 640x480 at 30 fps.[2]

[1] Any further information about the Kobuki base can be found in their official documentation found online at http://kobuki.yujinrobot.com/wiki/online-user-guide/.
[2] Further information on the specification of the camera can be found online at https://orbbec3d.com/product-astra/.

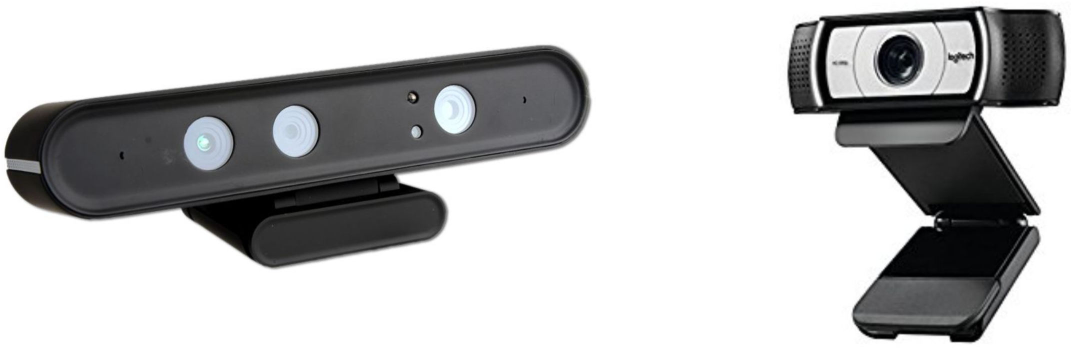

Figure 3.
Left: The Orbbec Astra 3D depth camera, mounted on the center platform of the turtlebot. *Right:* The Logitec C930e camera mounted on the top platform of the TurtleBots.

The human detection and recognition module requires high definition camera to maximize the working distance. For this a logitech C930e using resolution 1920 x 1080, using H.264 video format is used.

To control the system an Intel NUC computer is used. The NUC has an Intel Core i7 processor, uses 8GB DDR3 memory and an integrated graphics card. The specific NUC used here is the NUC5i7RYH.[3]

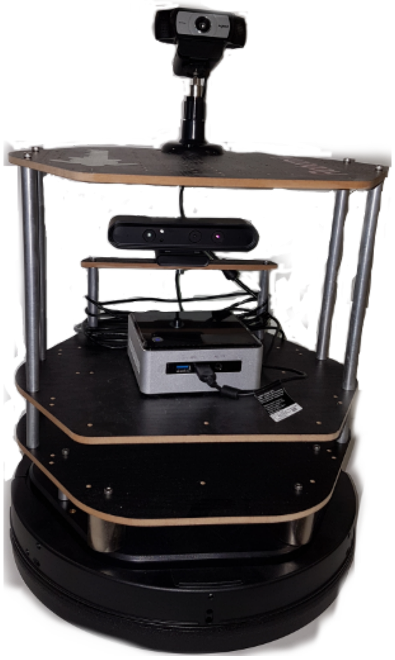

Figure 4.
Hardware setup of the TurtleBot 2. An Intel NUC computer is resting on the middle, above that is the 3D sensor (Astra Orbbec) and on top the Logitec RGB camera.

[3] Further information can be found online at https://ark.intel.com/products/87570/Intel-NUC-Kit-NUC5i7RYH.

DEMO-1 ROBOT SOFTWARE & ARCHITECTURE

The robots run identical software, but maintain a separate local current state and have separate IDs.

For Demo-1 each robot runs a Psyclone 2 system which contains a number of modules and catalogs. Underneath Psyclone the ROS system interfaces with the actual hardware sensors and motors.[4]

The components running in the Psyclone system relevant for Demo-1 are listed in Table 2 below. Catalogs can be seen as containers and arbitrators of data and modules are the processors, detectors and decision makers.

Table 2.
Main software components used in Demo-1.

COMPONENT	ROLE
CCMMaster **Type: CCMCatalog**	This is the central CCMCatalog which holds all the shared information in the whole system. Only one of these exists for each full system and each robot will connect to this via the network.
DemoRecording **Type: ReplayCatalog**	This catalog makes a recording of all the relevant messages in the system for later analysis of time and resources spent, timing of detections and decisions, etc. It takes no active part in the demo itself.
MessageDataCatalog **Type: MessageDataCatalog**	This catalog stores messages and their associated data for human viewing and debugging the system. It takes no active part in the demo itself.
PositionCollector1 **Type: CCMCollector**	This catalog collects local information about object (both robots and humans) and loads the information into the shared CCMCatalog. It will also allow querying based on time and space and allow the robots to negotiate about the position of objects in the scene.
RobotStatus	This module is the ROS system interface. It uses ROS to gather data from the robot sensors including the cameras and allows other modules to send commands to the robot such as navigation and turning.

[4] For more information about the Psyclone platform please refer to the following link: http://cmlabs.com/products

RobotSelf	This module analyses all the data gathered from the robot itself and converts this into the Psyclone data architecture. It also keeps the CCMCatalog up to date with the latest state, position, etc.
RobotNavigation	This module performs the search pattern negotiation via the CCMCatalog to agree with the other robots on where it should go next. It also allows a human operator to override the current navigation pattern and pauses the search pattern when the robot is currently tracking a human in the scene.
FaceRecognition	This module receives the video stream from the USB camera on the robot and analyses it for faces. For every face found it performs an identification as well as facial expression analysis.
HumanDetection	This module keeps track of the faces and humans detected in the scene and from a variety of data in the system it attempt to match the face with a body and/or legs and from this and its own position and orientation will calculate the actual scene location of the human.
Others	Numerous other system components have been developed that are fundamental (navigation, motor control, etc.) and not detailed here for brevity sake or because they are not essential for Demo-1.

The robots communicate via the CCMCatalog, a component explicitly designed to handle direct robot-to-robot communication and negotiation. At this stage the CCMCatalog is used to share information on humans that have been detected. The CCMCatalog takes no active part in the robots decision making as this is done independently by each robot – instead it acts as a centralised storage for observations and as a way that the robots can negotiate with each other about sub-tasks such as where a human is located, and where each should navigate next to ensure best observation coverage.

To update the CCMCatalog each robot has a CCMCollector – a module that collects the relevant data and communicates with the CCMCatalog. All observations of humans detected in the scene are continuously updated to the CCMCatalog by the CCMCollector. Each observation is tagged with metadata: (a) who made the observation, (b) when, (c) where and (d) the confidence of the correctness of the observation. Each robot can query the CCMCatalog for all such metadata.

MEASUREMENTS

In human-robot interaction it is ultimately the whole experience that matters to the end-user. The overall experience is impacted by the performance and coherent interaction of the whole system’s sub-components. In Demo-1 the ability of robots to interact with the real world and collaborate - as well as their efficiency in doing so - is our target for development. For this both sub-components and the overall system performance needs to be calculated. We use a mixture of sub-component measurements and overall performance measurements to provide an overall picture of the system at this stage of development (see Table 2 below).

The efficiency of the system as a whole as well as its speed will be measured through the use and measurement of CPU usage; for individual tasks as well as for the total time allocated for a chain of tasks related to the same goal. We also ran tests to measure the efficiency of detecting humans. This has the added benefit of serving as a baseline for comparison with future demonstrations, where the goal is to improve the accuracy and the speed of the detection over the course of the project.

Efficiency

What we call “efficiency” in Demo-1 is a multi-dimensional measurement of component and whole-system evaluation measures. The full set is listed in Table 3 below.

Table 3.
Overview of measurements used in Demo-1. All measurements are averaged over 6 runs during which data for the above measurements are collected.

Measurement Name	**Estimation of ...**	**Measurement Method**
Speed	internal processing speed (architecture).	Time difference between event start and timestamp of success message.
Effort	efficiency of the system as a whole.	Accumulated CPU seconds / Speed
Success Rate	how often the recognition works as planned.	The number of correct success message vs all reported messages.
Error Rate	how often the recognition works as planned.	Number of times the success message is wrong vs. / total success messages.
Wasted Effort	processing with incorrect conclusions.	%CPU used in producing incorrect conclusions (false positives + false negatives)

Human Detected	the time it takes a robot to know there is a human in the scene.	Wall-clock time: Timestamp (msec) of “human detected” posting minus the timestamp marking when the human entered a robot’s visual image (ground truth - timestamp generated manually by a human observer).
Person Recognized	the time it takes a robot to find the identity of a person that has been detected as a human.	Wall-clock time: Interval (in msecs) between the time a human is detected until a robot correctly posts his/her identity.
Human Recognition (collaborative)	the time it takes two robots in collaboration to find the identity of a person that has been detected as a human.	Wall-clock time: With both robots present, measured from the time a human enters either robot’s camera frame (timestamp generated manually by a human observer), to the time the person's identity is logged in the shared data structure (CCMCatalog).
Human Leaves	the time it takes a robot to record that a human identified as such has left its current visual frame.	Wall-clock time: Measured from the time the human leaves the scene (ground truth) until either robot posts "human left".
Human Leaves (collaborative)	the time it takes two robots in collaboration to record that a human identified as such has left either’s current visual frame.	Wall-clock time: With both robots present, the timestamp (msec) of a “human leaves” event logging in the shared data structure (CCMCatalog) minus the timestamp of the human leaving the area where robots can detect humans (generated manually by a human observer).
Search (collaborative)	the ability and time taken by two robots to negotiate where to go next during a visual search sub-task.	Wall-clock time: With both robots present, timestamp (msec) of completion of successful where-to-go-next negotiation (logged in the CCMCatalog) minus the timestamp of when the robot decided that it needed to move (also logged in CCMCatalog).
Visual Coverage	the amount of visual coverage a robot is able to perform.	Estimated from an estimated of the average movement of a robot during each demo session.
Visual Coverage (collaborative)	the amount of visual coverage the robots can achieve via collaboration.	$(VC_{r1} + VC_{r2}) - V_{overlap}$, where VC_{r1} is visual coverage provided by camera of robot 1, VC_{r2} is visual coverage provided by camera of robot 2, and $V_{overlap}$ is the amount of overlap between the two areas.

Visual Coverage

Since both robots search for humans and since they can communicate their findings via the CCMCatalog coordination of the visual area covered could make visual search more efficient, compared to each robot doing its own independent search. The aim is for them to reduce or eliminate overlap which would reduce both the effort and time spent on the task.

We define the visual coverage of a robot as the total area, within the area chosen for the demonstration, where if a human is standing, they should be recognized by the robot. Due to the characteristics of the robots, this area is defined as a cone of radius 2 meters and angle 30 degrees centered around the direction where the camera of the robot is aiming at.

The total available area for the robots to be covered is a rectangle of dimensions 3 x 10 meters and any overlap of the robots' vision cone with the area outside this is not counted for the evaluation of the visual coverage. With that in mind, if the robot's cone of vision is entirely located within the demonstration area, its visual coverage is a cone of 2D surface measuring 1.05 square meters total.

Total area covered V_{tot} is $V_{tot} = VC_{r1} + VC_{r2} - V_{overlap}$, where VC_{rn} is the area covered by one robot's camera and $V_{overlap}$ the area covered by both robots at the same time. In the ideal case scenario, $V_{overlap} = 0$. We estimate the total area covered, and the overlap, in the following way.

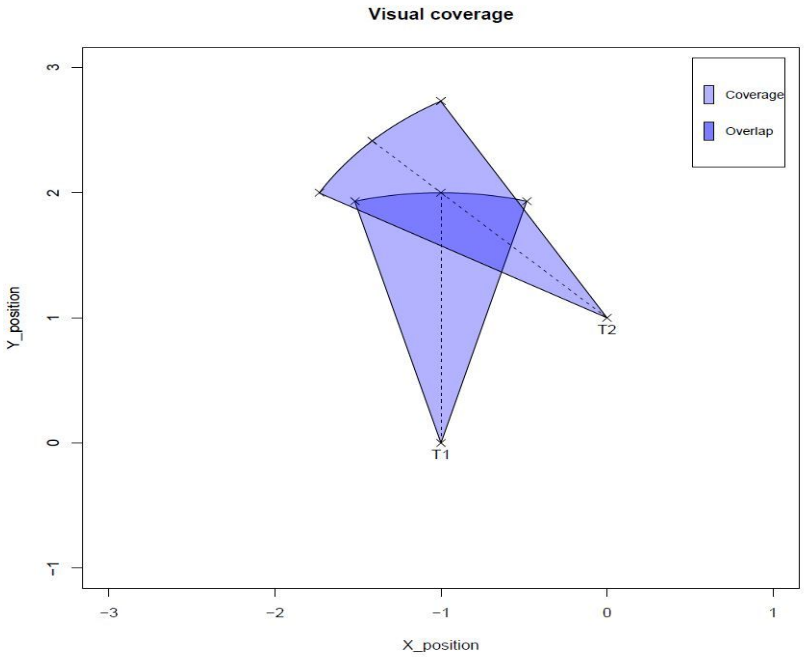

Fig 5.
Visual coverage and overlap between two robots' cameras

EXPERIMENTAL PROCEDURE & EXECUTION

The canonical demo run consist of the robots collaborating on finding humans, and at least 4 different people stepping into the scene to see if the robots recognize them. Detecting humans and identifying them was conducted the same way in all runs: the human walks in front of the robot at a distance of between 1-2 meters. Then they keep motionless for a total duration of 20 seconds, and then exists the test area.

During each run the robots collaborate via the CCMCatalog to share information about humans and to negotiate a search path when no human is present. If no human is present each robot follows the negotiated search path and when a human has been observed, using the facial recognition module from the 2D high resolution camera, the physical location of the human is computed by identifying either the associated torso or legs in the 3D depth sensor data. Using this information the human's location can be calculated taking into account the robot's own location and orientation.

A human's calculated location is entered into the CCMCatalog and the creation of a new object in the CCMCatalog triggers a notification of this event in the other robot. Once either robot has a human in their visual image they will collaborate to change their search pattern so that the human is tracked by one robot while the other robot searches other areas of the visual scene.

Once the human has left the scene (is no longer observed) the human will be marked as having left the scene and the robots resume their negotiated search pattern.

Each run had at least 4 human detection attempts.

EXPERIMENTAL RUNS

To ensure that all measurements were accurate and to fix any anomaly in the experimental setup three pilot runs were conducted on three separate days, before the final runs producing the final data presented here were executed. For each run, the robots always start in the same position and orientation. Each run lasted exactly 10 minutes during which a human enters the visual frame of either robot at least 4 times. The demo run with the largest number of detecting humans events had a total of 8 occurrences of attempted human detection.

RESULTS

The Demo-1 data shows that the system works as a whole fairly reliably with robots running hours at a time. The data also shows that all functions are in the right ballpark although some improvements are needed, especially in the computer vision and camera setup.

The main results are summarized in Table 4 below.

Table 4.

Summary of results. (Rows: See text below for short description of each measure; more detail is provided in Measurements section, above.)

EVENT	Speed (msec)	Effort (% CPU)	Success rate	Error rate	Wasted Effort
Human Detected	2978	55%	35%	65%	35%
Person Identified	3556	95%	25%***	75%	85%
Human Leaves	5181*	30%	80%	20%	12%
Search (Collab)	11587**	28%	87%	13%	2%

* based on a timeout setting after the human tracking was lost

** includes negotiation and navigation time

*** When a human is detected each frame of subsequent video is analyzed to identify the person; on average a correct ID requires 17 attempts, taking 2.03 seconds each, at 30 fps. We believe better cameras can improve this significantly; we are also looking into other approaches to improve on this point.

Speed: Time difference between event start and timestamp of “success” message
Effort: N of accumulated CPU seconds over Speed
Success Rate: Number of times “success” message is correct over total success messages
Error Rate: Number of times “success” message is incorrect over total success messages
Wasted Effort: %CPU processing with incorrect conclusions (false positives)
Human Detected: Interval between timestamp of “human detected” posting minus the timestamp marking when the human enters the area where the robots can detect humans.
Person Recognised: Interval in msec between timestamp of “human identified” posting minus the timestamp of the “human detected” posting.
Human Recognition (Collab): Interval in msec between timestamp when the person’s identity is stored in the CCMCatalog minus the timestamp of when the “human detected” posting.
Human Leaves: Measured from the time the human leaves the scene (ground truth) until either robot posts "human left".
Human Leaves (Collab): Measured from the time the human leaves the scene (ground truth) until the event is logged in the shared data structure (CCMCatalog)
Search (Collab): Measured from the time a robot decides it is time for it to move until the robot has successfully negotiated where to go via the CCMCatalog.

Detecting Humans

While detection works some of the time (35%), the data for the three humans tested here shows that the the current setup leaves something to be desired. To get the face detection to work reliably a face would have to be to be no more than 60 cm from the camera, meaning the humans needed to bend down to get recognized. However, even in this approach the system was only able to get 35% success rate. Partly this is due to how low the camera sits; when people stand at the comfortable distance of 1-2 meters the bright fluorescent ceiling lighting and the angle of the camera means faces are backlit, the aperture closes down making the faces too dark for the detection algorithm.

We are evaluating several ideas for improving the system on this aspect, none of which require extensive coding or excessive expenses, and we are hopeful that this aspect can be improved to acceptable/usable levels. It was observed that having the USB camera on top of the TurtleBot required the angle of capture to observe the ceiling lights, this greatly affected the lighting quality of the captured image and reduced the probability of recognizing a person correctly. One idea is to extend the height of the USB camera used for person detection and cropping the image for faster recognition.

Identifying Humans

Once a human was detected 71% of the people were identified correctly. But since the detection rate was low (35%) only 25% of the people in the scene were correctly identified.

In Demo-1 large high quality images are sent over the wifi network to separate computer performing the facial analysis. This creates a streaming bottleneck. To mitigate the issue a facial cropping method is being tested on the robot that takes image from the USB camera, crops faces if available, and sends the subset to face server for further analysis. Reducing the in-air data volume and workload on the face server is likely to improve ratio of correct recognitions over incorrect ones.

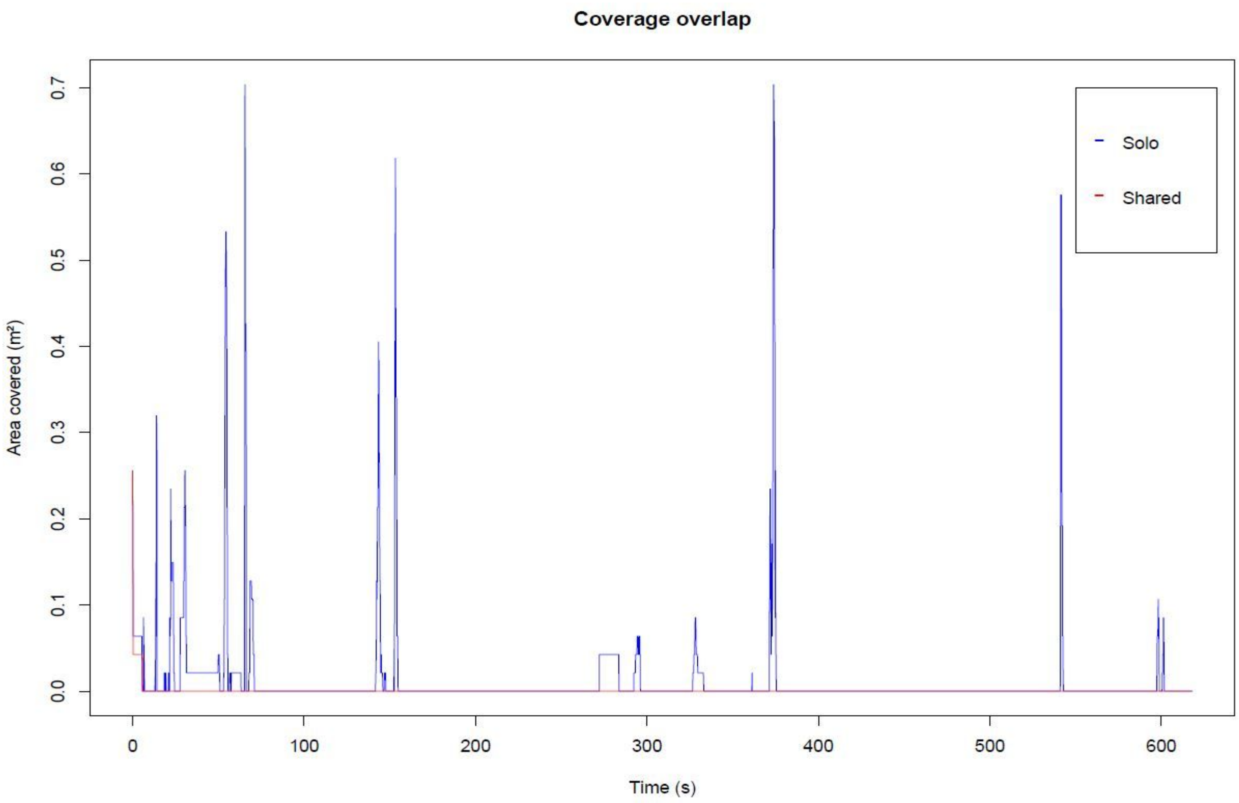

Fig 3.
Coverage overlap during the two test runs.

Visual Coverage

The data shows that improvements in coverage are clearly achieved when the robots collaborate on visual search.

When the two robots are not working together, the graph above highlights the fact that they will in fact frequently find themselves in the same vicinity during the test run. When collaborating, however (after the spike at the beginning due to the robots starting in the same area), the robots coordinate their visual surveillance of the area and are quite successful (albeit not perfect) at covering non-overlapping areas.

The average instant coverage overlap during the competitive run (“solo”) is of 0.01132 square meters. In the collaborative run (“shared”), the average instant coverage overlap is of 0.00038 square meters. In total, using the collaborative algorithms amounts for a decrease of the visual coverage overlap by 96%.

DISCUSSION & FUTURE WORK

The system development and implementation has reached a milestone for reliability and functionality, including integration of key components for future work (Demo-2 and Demo-3).

New tasks include improving the computer vision by testing a new camera and improving runtime efficiency and methodology in that area, including adding a custom physical extension (a ‘neck’) to the way the RGB camera is attached to the TurtleBot, as well as newer and higher quality cameras which handle lighting variations better.

Other steps have been identified as part of Demo-2 and include, on the software side, turn-taking and dialog control, speech recognition and synthesis, and on the hardware side microphones and speakers to support such interaction.