CoCoMaps Project

CMLabs | IIIM

CoCoMaps Demo-2 Report

Demonstration date: 6 March 2018

Link to video on Youtube: https://youtu.be/FsFgwil0or4

INTRODUCTION

The CoCoMaps Project is a joint effort by CMLabs (UK) and the Icelandic Institute for Intelligent Machines (Iceland) to integrate an architecture for robot control and interaction with humans using natural communication.

This report describes data from Demo-2 of the CoCoMaps project.

The goal of Demo-2 is to demonstrate the advances made in the development of the Collaborative Cognitive Map Architecture *final version*, including task collaboration, negotiation skills, and more. It is listed as Milestone 6 in the CoCoMaps including the following deliverables:

- **T8.D3** Final Collaborative Cognitive Map (Complete)
- **T10.D1** Demo 2: Collaborative Visual Search

This report describes the successful conclusion of Demo-2. It includes data showing how the robots collaborate, negotiate roles and tasks and interact with humans, in concert with other functionalities of CoCoMaps (including collaborative visual search, human detection and recognition and navigation that were demonstrated in Demo-1) integrated into a running system involving two robots with a human present in the robots' area of operation.

The results of processing times, CPU loads, and overall architecture reliability are shown to be within target ranges, providing a foundation for continuing onto the next steps of the project.

Demo-2 Goals. Demo-2 aims at demonstrating collaboration capabilities using dialog integrated with navigation and appropriate visual competencies as well as virtual control panel interaction, where two robots work in an environment receiving help from humans on their task. Specifically, the robots collaborate and communicate with each other and with a human, to perform a task initiated by the robots. The collaboration involves negotiation of roles, communication about their observations and dialogue acts that are both time- and context-dependent.

- Role negotiation
- Task negotiation
- Turn-Taking

We test this by running scenarios that test key aspects of these capabilities. To ensure consistency and data reliability we use task-driven scenarios that are run several times in the same area. To evaluate the collaborative aspects each run of the scenario is run with the robots in a "singleton" 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-2 is to demonstrate the successful design and implementation of the CCM to facilitate collaboration between robots and humans.

Table 1.
KPIs from CoCoMaps proposal relevant to Demo-2.

1	Ability of current state of the art running on one computer	M10		One computer able to see, listen and speak in simple setup	Video recording, statistics graphs
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
3	Ability of real-world multi-robot-human interaction using collaborative CMA and speech	M15	Two Turtlebots able to communicate via CMA	Two Turtlebots able to communicate with one human via CMA and voice	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 Demo 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 Demo 1	Measure added efficiency (speed, effort, error rate) of collaborative tracking
6	Efficiency of collaborative informatio n extraction through dialogue	M17	Initial measurement of extraction efficiency at current SOA implementation	Measurement of extraction efficiency at Demo 2	Measure added efficiency (speed, effort, error rate) of collaborative extraction
9	Human-leg and torso tracker using 3D information from the navigation camera	M17	Measurement of tracking efficiency at Demo 1	Measurement of tracking efficiency at Demo 2	Measure added efficiency (speed, effort, error rate) of collaborative tracking
10	Participant Negotiation Module, distributed reasoning/data fusion system for estimation of observations of the participants.	M17	Measurement of collaborative data sharing at Demo 1	Measurement of collaborative data sharing at Demo 2	Measure added efficiency (speed, effort, error rate) of collaborative data sharing

The report is organised as follows: First we describe the *Experimental Setup*, then we present *Results of Demo-2* based on figures collected from (multiple runs of similar) scenarios relevant to key performance indicators (KPIs).

KPIs from Demo-1 are relevant here and are included in Table 1.

EXPERIMENTAL SETUP

This section provides a short 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 6 meters. The lighting consists of built-in overhead fluorescent lights. The local Wi-Fi network provided communication between the robots and the base computers. The experimental setup for CoCoMaps Demo-2 was very similar to Demo-1. Two control panels were arranged at one end of the space, with approx. 1.5 meters between them with which the robots interacted virtually (as they have no arms and hands).

Demo-2 Robot Hardware

We use two identical TurtleBot2 robots[1] identical to those in Demo-1 except in that they are equipped with a better RGB camera, used for human detection and recognition, sitting on a new custom stand that raises it higher from the robot base, to better avoid glare from the overhead fluorescent lighting. The new camera is a Logitech BRIO with a resolution of 1920 x 1080 pixels, using raw uncompressed video, sufficient for the human detection and recognition module, which requires high definition camera to support increased working distances for face recognition.

The main computer is as before an Intel NUC, placed onto each TurtleBot structure.[2]

Figure 2.

TurtleBot 2 with the *Kobuki* base, including an *Astra Orbbec 3D* depth camera and an *Intel NUC* control computer. The *Logitech BRIO* USB camera on a stand, which also includes the *Jabra Speak* integrated microphone and speaker.

[1] TurtleBot 2 is an open-source hardware project built on the mobile Kobuki (http://kobuki.yujinrobot.com/wiki/online-user-guide/) 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. 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. 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. See http://www.turtlebot.com/turtlebot2/.

[2] The specific NUC used is the NUC5i7RYH. It has an Intel Core i7 processor, uses 8GB DDR3 memory, an integrated graphics card and Wi-Fi. Further information: https://ark.intel.com/products/87570/Intel-NUC-Kit-NUC5i7RYH.

For navigation, mapping and localizing a 3D depth camera, Astra Orbbec, is placed in the centre 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.[3]

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

DEMO-2 Robot Software & Architecture

As in Demo-1, the robots run identical software, but maintain a separate local current state and have separate IDs. As before, 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-2 are listed in Table 2 below. Catalogs can be seen as containers and arbitrators of data while modules are the processors, detectors and decision makers.

The robots communicate via the CCMCatalog (as in Demo-1). At this stage the CCMCatalog is used to share information on humans that have been detected. All robot decisions are made independently by each robot – the CCMCatalog acting as a centralised storage for observations, providing a virtual channel for the robots to negotiate with each other about sub-tasks including where a human is located, where each should navigate next to ensure best observation coverage, and their own position in the scene.

To update the CCMCatalog each robot has a separate CCMCollector module that collects 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.

[3] See https://orbbec3d.com/product-astra/.
[4] More information: http://cmlabs.com/products

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

COMPONENT	ROLE
CCMMaster Type: CCMCatalog	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 connects to this via the network.
DemoRecording Type: ReplayCatalog	Catalog that 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 Type: Psyclone core module	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.
RobotSelf Type: CCMCollector	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 Type: CCMCollector	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 Type: CCMCollector	Module that 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 Type: CCMCollector	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.

Table 3.
New software components used in Demo-2.

FaceFinder	Module for finding faces in each video frame.
RobotSelf	Module that collects all data relevant to the robot, including its position, orientation, identity, and current role.
SpeakerOutput	Receives text to be spoken and plays it; manages pausing audio (during hesitations), flushing speech output buffer.
RobotSpeechMonitor	Keeps track of which robot is speaking when.
StopSpeakingDetector	Special high-speed detector for managing stops and starts during multi-party dialogue.
SpeechRecogniser	This is the front-end module to the Nuance speech recognizer that interfaces with the Psyclone system. Receives recognition packets from Nuance and posts as Psyclone messages.
OverlapDetector	Dedicated module for detecting when overlaps in speech occur. Used by robots to flush speech recognition buffer to clear misrecognitions (guaranteed to be faulty during overlapping speech).
InterruptionDetector	Detects interruptions. Used by Turn-taking module, TaskDialogManager, and others.
TaskDialogManager	Manages the interaction with humans and tracks the state of tasks that the robots are engaged in. The TDM handles context-dependent interpretation of actions and speech acts (commands, requests, etc.), manages task progress and robot task division of labour. Manages task and sub-task navigation using a task tree.
MeaningExtractor	Responsible for turning the user's behaviour into context-sensitive responses. Receives text (and, in future, gestures, facial expressions, and more), parses it, maps it into reified 'meaning structures' that are used to compose response (real-world action and/or dialog act).
RoleNegotiator	Responsible for negotiating either shared or exclusive roles for the robots.
TaskNegotiator	Responsible for negotiating which of the robots should carry out a task, based on their current roles and other parameters.
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-2.

MEASUREMENTS & METHODOLOGY

In human-robot interaction it is ultimately the whole overall experience that matters to the end-user. The overall experience is impacted by the performance and coherent interaction of (most or all of) the system's sub-components.

In Demo-1 the ability of robots to interact with the real world was our target for development.

In Demo-2 we add the ability to interact in groups (two robots, one human) using language.

Variables & Measurements

We measured a number of variables over a series of similar scenarios. Here is an account of these, broken down by measurement type.

Table 4.
Measurement types used in Demo-2.

Measurement Name	Measurement of ...	Measurement Method
Speed	Average of internal processing speed (architecture).	Time difference between event start and timestamp of success message. Using messages produced by relevant modules and recorded in CoCoMaps catalogs. Based on a minimum of 10 trials.
SD	Standard deviation	$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$
Min	Lowest value recorded in the trials.	
Max	Highest value recorded in the trials.	

Table 5.
Variables measured in Demo-1 and Demo-2.

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 Identified	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 Identified (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 Detected

Measurements for how well humans are detected relies on the chain HumanEnters → FaceFound. This covers the initial detection of the human, tracking is done once the human has been identified.

Person Identified

We use the same labels as for the HumanDetected part of the chain (see above), with the message type HumanAppearedSelf, which is posted by the HumanDetection module, once the location in 3D has been determined using the body and leg detector.

Human Identified (Collaborative)

Once a human has been detected by one robot it will notify the other robots via the CCMCatalog. This uses the same negotiation mechanism as regular observations.

Human Leaves

To measure the robots' ability to detect when a human has left we search through the logs for the HumanLeft message, which is posted by the vision system, take note of the timestamp (clocks are synchronized across the CoCoMaps architecture) and subtract from this the timestamp of a manually marked signal in the log files for when the human actually left the image.

Table 6.
Overview of new measurements used in Demo-2. The higher each of these are, the less artificial – i.e. more natural – the interaction is.

Measurement	Estimation of ...	Measurement Method
Speech-to-Text	Correctness of transforming audio stream to the correct words.	Percent correct of words for sentences in Demo-2 and Demo-3.
Turn-taking	The ability and time taken by a communicating robot to detect turn-taking events.	Wall-clock time: The time between a human giving turn, based on microphone signal, and the internal turn-taking state machine posting a message to the whole system to that effect.

Speech-to-Text

Speech-to-text is the transformation of audio signal to words. To get a baseline for dialogue understanding (see *Dialogue Understanding* below) this is an important measure because, since understanding relies heavily (but not only) on the speech output of the humans, the quality of the transformation places a ceiling on how well dialogue understanding can work.

We tested 30 sentences similar to those that are used in a typical interaction in Demo-2 and measured the number of words that were correctly transcribed to text. The speaker was a native speaker of French. The Nuance speech recognizer relies on an American pronunciation library and language model, which is not optimal for users with a foreign accent.[5]

Turn-Taking Smoothness

To evaluate the smoothness of the interaction, one measure is the quality of the unfolding turn-taking. In Demo-2 the robot always has something to respond to when the human gives the turn, and the immediacy of taking the turn is a measure of smoothness.

Tests were conducted with multiple occurrences of the human speaker giving turn to the robot, to measure how accurately the robot does take the turn.

Instances of human "giving turn" are symbolized in the pub-sub system by messages of type "OtherGivesTurn", produced from lower-level signals including speech and vision. If properly detected by the robot, and decided to act upon, the robot outputs the message type "IAcceptTurn". In order to measure the ability of the robot to react accordingly to human reactions, we split the dataset, looking at the succession of events that predated a "IAcceptTurn" event, using system timestamps for estimating latency between the relevant messages.

We consider that any cue of type "OtherGivesTurn" that has *not* been acted upon within 2,5 sec is lost, meaning the robot has failed to take turn. In addition, if several cues of "OtherGivesTurn" are given within the three minutes before the robot decides to take turn, all

[5] In spite of repeated attempts at getting different language models for the speech recognizer, Nuance was not able to fulfill this promise according to the description and spec for their recognizer. Since much of the quality of the interaction hangs on the speech recognition working well, switching to a different speech recognizer is therefore high on the priority list for low-hanging fruit for improving the system.

but one of these cues are effectively wasted. This enables us to measure the average "wasted time" in our turn-taking system.

Experimental Execution

The demo consists of the following: Two idle robots in an 3x6 meter area populated by two (virtual) control panels.[6] Whenever a human enters a scene they request the human's help for performing a sequential task involving one of the two panels. The robots already know the steps need to perform the task, but they require oversight from a human, which they receive via natural language. This scenario was repeated several times to produce reliable measurements for each of the target variables on the relevant dimensions, as reported in the table below.

During each run of the task the robots collaborate via the CCMCatalog to share information about humans and to negotiate roles when a task has been identified. If no human is present each robot follows the negotiated search path, as in Demo-1. When a human is observed the robots request assistance with a task they know how to perform.

To ensure that all measurements were accurate and to fix any anomalies in the experimental setup, several runs of the scenario were performed. Each run lasted approximately 10 minutes.

[6] The panels are displayed on a screen with which the robots interact via wireless messages.

RESULTS

Demo-2 data shows that the expansion of the system has not decreased reliability of its operation; as before the robots run hours at a time. While there is clear room for improvement on many measurements, it also shows that target functions perform numerically in the right ballpark.

The main results are summarized in Tables 7 and 8 below; Table 7 includes measurements from Demo-1 for comparison; Table 8 presents new measurements for Demo-2.

Table 7.
Summary of Demo-2 results for repeated measurements of Demo-1. Numbers (in parenthesis) from Demo-1 are included for comparison.

EVENT	**Success rate** (%)	**Speed** (msec)	**SD**	**Min**	**Max**
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	78 (35)	870 (2978)	820	40	2800
Person Identified Interval between timestamp of "human identified" message minus the timestamp of the "human detected" message	81 (25)	8340 (3556)	7170	920	23160
Person Identified: Collab. Interval between timestamp when the person's identity is stored in the CCMCatalog minus the timestamp of "human detected" message	81 (---)	1680 (---)	770	490	2810
Human Leaves Measured from the time the human leaves the scene (ground truth) until either robot posts message "human left"	100 (80)	630 (5181)	394	1410	13990

Detecting Humans

Having the USB camera on top on an extended pole has greatly reduced some of the negative effects from ceiling lighting on the quality of the captured image and the ability to detect a person. Cropping the image has resulted in faster processing.

Identifying Humans

The method used for person identification performance in Demo-1 showed that for a larger recognition area, a greater distance to the camera, and less sensitivity and interference due to lighting, were needed to improve performance. On this basis we made three main changes: A better camera, better camera placement, and more intelligent management of wireless bandwidth for image transmission. In Demo-1 large high quality images were sent

over the Wi-Fi network to separate computer performing the facial analysis. This created a streaming bottleneck. To mitigate the issue in Demo-2 a facial cropping method was developed that takes image from the USB camera on the robot, crops any faces found, and sends only the subset to the face server for analysis. Reducing in-air data volume and workload on the face server improved ratio of correct recognitions over incorrect ones by speeding up the process.

In light of the overall goal of naturalness we made the choice to have more accurate algorithms, although this improved accuracy is at the cost of speed the processing time is nevertheless within acceptable margins for using this information during the interaction, e.g. to mention a person's name during a conversation. Being capable of recognizing humans without having them needing to bend over and keep still for 5 seconds is a significant improvement.

Compared to the initial version in Demo-1, the system in Demo-2 is capable of:

- detecting humans from a longer distance
- detecting humans with a better overall accuracy

Table 8.
Summary of new Demo-2 measurement results.

EVENT	**Success rate** (%)	**Speed Ave.** (msecs)	**SD**	**Useful time** (msecs)	**SD**	**Wasted effort**	**SD**
Turn-Taking Smoothness % turns with no overlaps and <2,5 sec pauses between turns	97	2120	362	182	830	492	363
Role Negotiation Time and effort measurement from one robot deciding that a role needs to be assigned until the negotiation has been completed.	100	0.42	0.087	0.42	-	0	-
Task Negotiation Time and effort from one robot deciding that a task needs to be carried until the negotiation has been completed about which robot has accepted the task.	100	0.068	0.038	0.42	-	0	-

Turn-Taking Smoothness

We measure the efficiency of the Turn-Taking system by the average time difference between the reception of the human speech input and the decision taken by the system to act upon it.

- Proportion of turn-taking events one or more wasted cues: 31%
- Proportion of failed turn-taking events (robot did not take turn): 25.6%
- Success rate (robot took turn): 74.4%

Role Negotiation

As the numbers indicate, Role Negotiation (which robot is 'communicator' and which one is 'task executor') is a fast and seemingly bug-free operation. The negotiation mechanism, and their supporting processes, operate very reliably and efficiently.

Task Negotiation

The same can be said of task negotiation as of role negotiation, which in large part relies on the same mechanisms (but not entirely). Negotiation of tasks happens internally to the robots, deciding which one has to accomplish which given task.

Table 9.
Summary of new Demo-2 measurement results: Speech-to-text.

Measure	Ave % Correct	Description of Measure
Speech-to-Text % correctly transcribed words from speech	66	Average of words that are transcribed correctly by the speech recognizer (Nuance) during dialogue.

Speech-to-Text

The Nuance embedded speech recognition software being used for this project is only supplied with American accent support. This is a known source of problems for recognition within the English accents and typically introduces an additional error rate of between 15-25% for speakers of non-American origin.