DOI: http://dx.doi.org/10.7551/978-0-262-32621-6-ch045

Associating Nearby Robots to their Voices

Plamen Ivanov and Dylan Shell
Texas A&M University, College Station, TX 77840
flame09@tamu.edu

Abstract

It can be useful for a robot in a team to recruit the help of others after considering their poses in space. When the initiating robot is aware of the network addresses (or IDs) of the robots it may ask for cooperation over the network. But if the initiating robot is ignorant of their IDs then it becomes rather more complex or time consuming. In this paper, robots use a simple visual (locally sensed) gesture (such as a light being turned on or off) in addition to wireless messages to quickly and effectively establish a one-to-one association between physical location IDs and wireless IDs. This paper identifies, describes, formalizes, and then generalizes the problem of establishing a local association between neighboring robots' physical location IDs (in the reference frame of an observing robot) and their wireless IDs. We describe three algorithms that solve this problem: two are deterministic algorithms and one is a probabilistic algorithm. Using the algorithms and formalism, we examine the structure underlying this association problem.

Introduction

Robots cooperating in teams often benefit from spatially referenced communication: "*Would the robot to my left, please help me move this piano?*" But many current implementations are influenced by the fact that practical perception of other robots is sensor-based (*e.g.*, using vision via cameras) while communication is based on network technology (*e.g.*, Wi-Fi, radios). Since the same robot will have different identifiers in these two realms, an association must be formed if our robot is to request help over the network: "*Will robot 192.168.0.56 help move this piano?*" We consider the problem of associating two independent channels of communication to form a one-to-one matching between sources in one channel and communication sources in the other.

We specifically consider the question of relating a robot with some relative pose (*i.e.*, when the "where" of the robot is known in the local frame) to a name which can be used to refer to that robot directly (*i.e.*, the "who" of that robot). The robots comprising the system are assumed to be able to interact through two independent communication channels. Using only these two channels and no global localization knowledge or global communication, all robots in the system must each solve:

For each robot "visible" in the second channel, determine and associate the appropriate network address in the first channel.

The many different variants of this problem are usually addressed by using an array of visual fiducials and a hand-crafted list of associations (Garrido-Jurado et al., 2014; Cassinis and Tampalini, 2007; Howard et al., 2004), or by initially calibrating and identifying the robots. These solutions are *ad hoc* and focus on practical cases of a specific variant of the problem (Mathews et al., 2010).

In terms defined by Støy (2001), the first channel in the motivating system is a non-situated channel, such as a Wi-Fi or radio, and the second channel is a situated communication channel, such as a light source attached to a robot. The capabilities of the situated visual channel are assumed to be minimal: a simple on/off communication strategy over the visual channel is enough to complete the association successfully. This could be implemented by moving the robot (*e.g.*, performing a predefined wiggling repertoire), or switching on a light, or other such visually identifiable *gesture* (*e.g.*, an instance of a protocol on just such a channel is given in Dieudonné et al. (2009)). The non-situated radio channel is assumed to have a message size which can at minimum fit the radio ID of a robot.

In this paper, we identify and formalize the problem generally, showing that it has an interpretation from an information filtering perspective (LaValle, 2009). Our paper describes two deterministic algorithms and a probabilistic algorithm solving this association problem, each having distinct features and drawbacks. The formalism and solutions are applicable to many circumstances in which *indexical* knowledge must be related to *objective* knowledge (Agre and Chapman, 1987; Lespérance and Levesque, 1994) because the former represents task relevant information —natural in the one channel— while the latter permits targeted transmission —a straight-forward operation in the other channel.

The generalized association problem impinges on questions of task representation, addressing, communication, and localization — each topics that are subtle and nuanced in their own right. For example, although in the motivating piano moving scenario the indexical, task-relevant channel is the visual, situated channel, this need not be the case. Using this formalism we have made a step towards identifying and understanding the subtleties involved.

Related Work

Practically speaking, marker systems are limited in the number of unique identifiers which can be encoded, suffer from reliability issues of correctly identifying markers, have restrictions on the maximum distance at which markers can be identified and in what environments they can be deployed, and have significant hardware and software requirements (Garrido-Jurado et al., 2014; Cassinis and Tampalini, 2007). Solutions that require calibration of global knowledge are especially undesirable as system size increases.

The idea of *spatially targeted communication* in Mathews et al. (2010) and the model and solutions they use have similarities with the model and solutions we describe. However, their approach is only a solution for a pair of robots in the system. The model lends itself to analysis through the extinction probability of a branching processes (Haccou et al., 2007); we return to discuss this means of analysis later. Dieudonné et al. (2009) present a multi-robot system in which robots are indistinguishable from each other and there is no explicit means of communication. Their approach is remarkable because it considers a weak synchronicity model, but it makes strong sensing assumptions, precluding use in most real robot settings.

Mutual or collective localization in a multi-robot system (*e.g.*, Franchi et al. (2009); Fox et al. (2000)) is related to the association problem presented in this paper because a global metric reference frame provides unique labels for each robot. Oftentimes, however, the cost of localizing all the robots is greater than one wishes to pay merely for targeted communication.

Problem Definition and Formalism

In this section we give a precise description of the multi-robot system and the problem which we address.

Motivating System

Consider a multi-robot system composed of n robots, which are spread arbitrarily across an environment. Each robot in the system can directly observe and distinguish from the environment some nearby robots using a camera. Robots can also flash a light on and off. The camera and light form a single bit communication channel — the **camera channel**. Each robot can also directly communicate through messages over a **wireless channel** with robots in range. Each robot has a **wireless ID** and itself assigns **location IDs** (for example, local range and bearing pairs) to robots it can recognize through its camera.

Every robot starts knowing n and the IDs of robots they can receive messages from in either of the channels. The robots have no knowledge of which wireless ID is assigned to which robot. The operation of the robots is modeled with synchronous, discrete time steps. Both channels have a limited transmission range *i.e.,* a robot is unable to receive messages from all robots in the system directly.

The goal is for each robot to form a one-to-one association between the location IDs of robots in the camera channel and the wireless IDs of robots in the wireless channel.

Generalized Formal Specification

A generalized formal definition of the problem follows. Graphs representing the system are depicted in Fig. 1.

Definition 1. *We model a multi-robot system that has one radio communication channel (channel 1) and one physically situated communication channel (channel 2) with the tuple:* $\langle G_1 = \langle V, E_1\rangle, G_2 = \langle V, E_2\rangle, C_1, C_2, f_1, f_2\rangle$ *where*

1. V *is a set of vertices, each representing a robot.*
2. $G_1 = \langle V, E_1\rangle$ *and* $G_2 = \langle V, E_2\rangle$ *are two directed graphs representing the connectivity of the system of robots over the two communication channels,*
3. $E_1 \subseteq V \times V$ *has directed edges s.t.* $e \equiv (v_i, v_j) \in E_1$*. This is interpreted as robot* i *being able to receive a one-hop message from robot* j *over channel* 1*,*
4. $E_2 \subseteq V \times V$ *is analogous, but for channel 2,*
5. $E_2 \subseteq E_1$ *– this is the channel* ***inclusion property****, stating that if* i *can receive a channel 2 message from a robot then it is guaranteed to be able to receive a channel 1 message from that robot too,*
6. f_1*,* f_2 *are labeling functions of the form* $f : V \times V \mapsto C$ *which are applied to channels 1 and 2, respectively.*
7. C_1 *and* C_2 *are the label sets for channels 1 and 2.*

Each channel has an associated labeling function and the set of labels correspond to addresses (wireless IDs and location IDs, respectively). In the motivating system, the labeling functions will provide unique IDs for the wireless channel (IP addressed) and locally unique IDs (defined precisely, next) for the location channel. The labeling functions are not known to the robots. The only requirement is that whatever procedure makes the ID assignment does not violate the following property:

Property 1. *Labeling functions have:*

$$\forall x, y, z \in V, e_y = (x, y) \in E_{ch} \wedge e_z = (x, z) \in E_{ch} \Rightarrow f_{ch}(e_y) \neq f_{ch}(e_z).$$

Property 1 ensures that each neighborhood will have non-repeating IDs, which we call *locally unique IDs*. In the motivating system, it applies to the location IDs. It also holds for global IDs which are not locally in conflict *i.e.,* no individual robot will see the same wireless ID broadcasted by two different robots in its wireless neighborhood. Note that any labeling that provides unique IDs satisfies property 1.

To examine an individual robot's perspective we introduce several additional concepts.

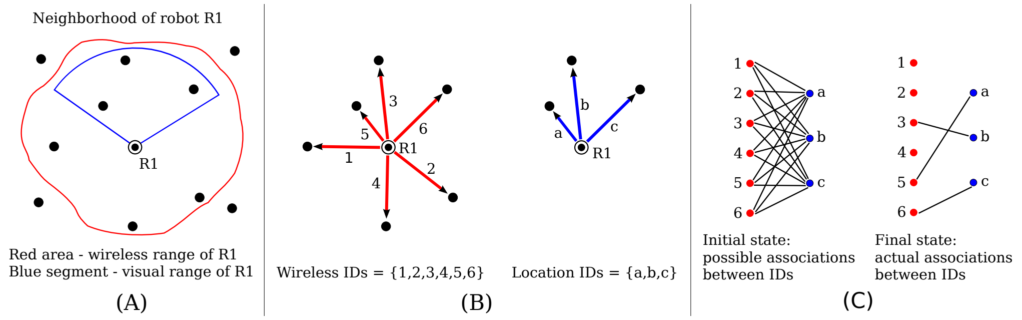


Figure 1: This figure presents a view of a local neighborhood around a robot in a multi-robot system (A) and some of the graphs which are associated with the channel association problem (B),(C). The wireless communication channel (in red) connects robot $R1$ to some of the robots surrounding it. The camera communication channel (in blue) has a more restricted view of robots in the system. Robot $R1$ is only aware of the set of IDs which correspond to the connection edges of each channel. The robot represents its knowledge of the association between IDs as a bipartite graph as shown in (C).

Definition 2. *A* ***neighborhood*** *is a subset of the system centered on an individual robot i. A neighborhood can be described by the tuple $\langle V^i, E_1^i, E_2^i, f_1, f_2, C_1^i, C_2^i\rangle$ where*

1. *i indicates the robot centered in the neighborhood,*
2. *V^i consists of robots for which a directed edge $e \equiv (v_i, v_j) \in E_1$ exists,*
3. *E_1^i and E_2^i contain all directed edges in E_1 and E_2 which have robot i as source.*
4. *C_1^i, C_2^i are the labels given to edges in E_1^i and E_2^i by f_1 and f_2, respectively.*

Definition 3. *The* ***local view*** *of robot i is the information directly available to the robot. It can be described by the tuple $\langle n, C_1^i, C_2^i, O^i = \langle O_1^i, O_2^i, ...O_t^i\rangle\rangle$ where*

1. *n is the maximum possible robots in the entire system,*
2. *C_1^i and C_2^i are label sets as previously defined,*
3. *$O^i = \langle O_1^i, O_2^i, ...O_t^i\rangle$ is a sequence of observations (Def. 5) of the robots in the neighborhood of i, and t is the time step during which an observation was made.*

Definition 4. *An* ***activation*** *captures the ground truth of the communication state of a neighborhood for a given time step t, described by the tuple $A_t^i = \langle C_{1At}^i, C_{2At}^i\rangle$ where*

1. *C_{1At}^i is the set of labels of robots which are in range of and from which a message was sent to robot i on channel 1 during time step t,*
2. *C_{2At}^i is defined similarly for channel 2.*

Definition 5. *An* ***observation*** *is an individual robot's perception of an activation for a given time step t and is described by the tuple $O_t^i = \langle C_{1Ot}^i, C_{2Ot}^i\rangle$ where*

1. *C_{1Ot}^i is the set of labels of robots from which a message was received by robot i on channel 1 during the time step,*
2. *C_{2Ot}^i is defined similarly for channel 2.*

Given the assumptions about no failure and no interruption in messages, activations and their corresponding observations are identical.

Definition 6. *Each robot maintains an* ***information state****, or* ***i-state****, which represents the current knowledge the robot has about its neighborhood. The i-state is described by the bipartite graph $S^i = \langle C_1^i, C_2^i, E_{1-2}\rangle$ where*

1. *C_1^i and C_2^i are label sets as previously defined and represent the two distinct vertex sets of the graph,*
2. *E_{1-2} is the edge set of the graph and represents possible associations between elements from each label set.*

Each robot starts in complete ignorance of the actual one-to-one association of labels. As observations arrive, it updates its i-state by removing edges from the graph.

Definition 7. *The* ***one-to-one association problem*** *for each robot i is:*
Input: a local view with an observation sequence,
Output: a complete or partial one-to-one association between channel 1 labels and channel 2 labels for the label tuple $\langle C_1^i, C_2^i\rangle$ corresponding to i's neighborhood.

Algorithms to Solve the Association Problem

In this section we present three algorithms that can be used to achieve the one-to-one association problem in a multi-robot system. The algorithms operate for a system with the following properties:

1. There are no communication failures, lost messages or message degradation.
2. The channel inclusion property holds.
3. The minimum message size for the wireless channel is $\log_2(n)$ bits.
4. The message size for the visual channel is 1 bit.
5. Every robot can receive and process messages from any number of robots in each channel during a given time step. Robots can only send one message in each channel during a given time step.

A **listener** is a robot from whose perspective we are analyzing the system. A **message pair** consists of the transmission of a message in each channel by a given robot during the same time step. The earlier assumption that each listener begins knowing the number of robots it can receive messages from in each channel can be achieved by having a single time step in which every robot transmits a message pair.

Naïve Algorithm.

In the Naïve algorithm, every robot emits a visual message during the time step which is equivalent to the robot's wireless ID. A listener simply fills a table associating a wireless ID to a location ID which is currently active. The pseudo code for this algorithm can be seen below.

Algorithm 1 Naïve Algorithm

```
procedure NAÏVE
    matchTable                        ▷ [wireless ID, location ID] pairs
    for t ← 1, n + 1 do               ▷ For each possible wireless ID
        if t − 1 == myID then
            broadcast(visualMessage)
        end if
        if messagesReceived() then
            if local_robot.isActive() then
                pair ← [t − 1, locationID]
                matchTable.add(pair)
            end if
        end if
    end for
end procedure
```

Logarithmic Algorithm

In the Logarithmic algorithm the fact that n identifiers can be represented with $\log_2(n)$ bits is used. The listener associates each observable robot with an array of size $\log_2(n)$. At time step t, all robots whose t^{th} bit is equal to 1 will emit a visual message. After each time step a listener can fill one more bit in the array of bits corresponding to the wireless IDs of the visually active robots. The pseudo code is below.

Algorithm 2 Logarithmic Algorithm

```
procedure LOGARITHMIC
    self_bit_arr ← to_bit_array(my_ID)
    table_arr ← array[num_neighbors][log2(n)]
    for t ← 1, log2(n) + 1 do
        if self_bit_arr[t − 1] == 1 then
            broadcast(visualMessage)
        end if
        for j ← 0, num_neighbors do
            if neighbor[j].isActive() then
                table_arr[t][j] = 1
            end if
        end for
    end for
end procedure
```

Probabilistic Algorithm

In the Probabilistic algorithm all robots construct an $h \times s$ i-state matrix to represent their i-state. Here h represents the number of robots in direct communication over channel 1 (C^i_1 from S^i) and s – the number of robots over channel 2 (C^i_2 from S^i). The i-state starts filled with 1's, which indicates complete ignorance of the one-to-one associations. Each cell in the i-state matrix corresponds to an edge in E_{1-2} from S^i. The wireless IDs (labels) of all robots that are within communication range are associated to an activity array (C^i_{1Ot} from O^i_t) that keeps track of which robot transmitted a message in the current time step (1 is used to denote a transmission). Each wireless ID is given an index $w \in [0, h-1]$. Similarly, the location IDs are associated with an activity array (C^i_{2Ot} from O^i_t). Each location ID is given an index $l \in [0, s-1]$. At any moment the i-state matrix contains the listener's current information about the matching between wireless IDs and location IDs.

During each time step, each robot emits a message pair with 50% probability. Every listener fills in an $h \times s$ observation matrix, which captures an observation of the neighborhood made during the current time step. This matrix is filled based on the following principle: The $[w][l]$ entry in the observation matrix is filled with a 1 if both the w^{th} wireless ID and l^{th} location ID were active during the current time step. All other entries are filled with 0's.

Once the observation matrix has been constructed it is used to update the i-state matrix of the listener. All $[w][l]$ entries from the observation matrix for which either the w^{th} wireless ID or l^{th} location ID have been active in the current time step are multiplied with the corresponding i-state matrix entries. This produces a new i-state matrix.

Once the i-state matrix contains only s 1's, the one-to-one association problem for the listener and its neighborhood is complete. The non-zero $[w][l]$ entries tell us that the w^{th} wireless ID and the l^{th} location ID belong to the same robot.

Pseudo code appears in **Algorithm 3**, and an i-state and observations for a single run of the algorithm for one neighborhood are depicted in Fig. 2.

Algorithm 3 Probabilistic Algorithm

```
procedure PROBABILISTIC
    broadcast(messagePair)
    h                                 ▷ number of wireless IDs in range
    s                                 ▷ number of location IDs in range
    istate[h][s] ← 1's
    observ[h][s]
    wifi_activity[h]                  ▷ 0 - inactive, 1 - active
    vis_activity[s]                   ▷ 0 - inactive, 1 - active
    for each time step do
        if fairCoinFlip() then
            broadcast(messagePair)
        end if
        wifi_activity ← receivedWifiMsgs()
        vis_activity ← receivedVisMsgs()
        observ ← fillObs(wifi_activity, vis_activity)
        istate ← updateState(observ)
        if istate.isFinal() then
            ready ← TRUE
        end if
    end for
end procedure
```

Information Space Interpretation

Although only the probabilistic algorithm explicitly uses the idea of an information space, both deterministic algorithms have an information space interpretation. In this section we examine the information space explored by the algorithms.

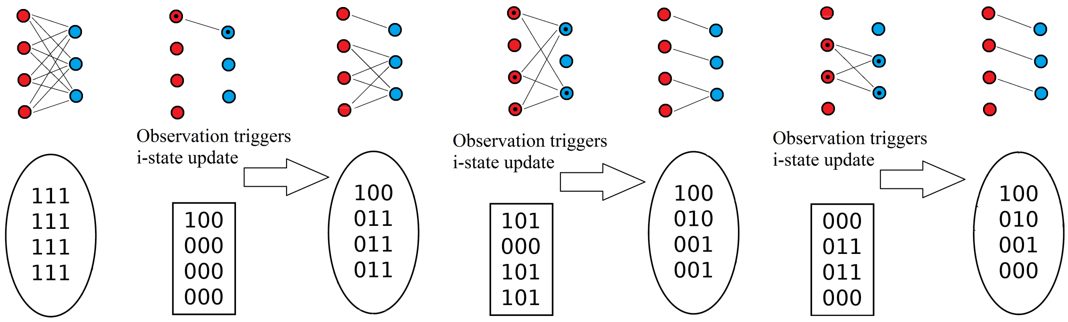


Figure 2: An i-state path through the transition graph. Here we can see the i-state of the robot (the matrices in ovals) and its progression as the robot makes observations (the matrices in squares). Bipartite graphs corresponding to each i-state and observation appear at the top.

Structure of Individual I-States and Observations

As observations about the neighborhood are made, a listener can update its i-state accordingly. The initial i-state is an $h \times s$ complete bipartite graph. An observation partitions the i-state bipartite graph into active and inactive vertices. All edges between an active and inactive vertex are eliminated from the i-state graph. The partitioned vertices form bipartite sub-graphs. Once such a partitioning occurs, the listener has reduced the potential associations between wireless IDs and location IDs. Every subsequent observation can split the already existing partitions further.

An observation can partition the vertices in such a way that it brings no additional information about the neighborhood. There are two observations which always bring no information: all robots are active or no robots are active. These are the zero-information observations. Observations can be paired based on symmetry – the partitioning these pairs induce on the i-state bipartite graph is the same.

The final i-state is a disconnected bipartite graph which is composed of bipartite sub-graphs of size 2 or 1. The sub-graphs of size 2 indicate a one-to-one association between a wireless ID and a location ID while the sub-graphs of size 1 show that a wireless ID belongs to a robot which cannot be observed by the listener.

The number of i-states for a given neighborhood is finite and can be found by calculating Bell's number for a set of size h. Bell's number counts the number of ways a set of a given size can be partitioned in.

The number of observations for a given neighborhood is 2^h. Since we have assumed an activation probability for a robot to be 50% (during the probabilistic algorithm), all neighborhood activations and corresponding observations have the same probability of occurring.

I-State Transition Graph

The i-states and observations for a given neighborhood form the **directed i-state transition graph**. The i-states form the vertex set of the transition graph and the directed edges correspond to the set of observations which lead from one i-state to another. A version of the transition graph can be seen in Fig. 3. The figure shows all i-states for a 4×3 neighborhood (4 wireless IDs and 3 location IDs), all i-state to i-state transitions, and the number of observation for each transition.

The graph highlights properties of the i-state space:

1. The i-states can be grouped into several categories based on the number of partitions of the initial i-state bipartite graph they contain. We call these categories **levels**. There are h levels in each transition graph corresponding to same size partitions of the graph S^i. The initial i-state is at level 1 (trivial partition of the graph S^i), and the final i-state is at level h (there are h partitions of the graph S^i). Any intermediate level will have between 2 and $h-1$ partitions of S^i.
2. At each next level the number of zero-information states doubles.
3. At each level the transition probability is equal for all transitions (for the probabilistic algorithm).
4. Each i-state can transition to itself (through zero-information observations).
5. There are no transitions between i-states from a higher partition level to a lower partition level and inside a level (with the exception of self-transitions).

Deterministic Algorithms and the I-State Graph

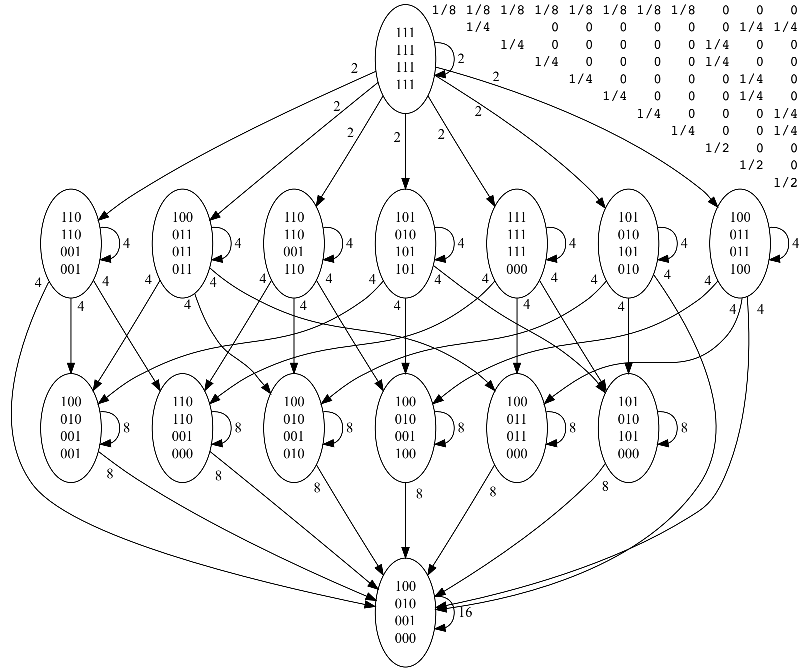
The Naïve algorithm uses only observations resulting from a single active robot. This restricts the paths through the transition graph to a subset of the 1-level transitions. The Logarithmic algorithm, on the other hand, allows any n-step path through the transition graph. The specific path for both algorithms is predetermined once a neighborhood is formed.

Algorithm Performance Analysis

After examining the information space we can examine the performance of the three algorithms presented earlier.

Naïve Algorithm

The Naïve algorithm runs for n time steps, where n is the number of unique IDs that can be used by the system of robots. The algorithm runs for n time steps regardless of how many of the n identifiers are in use by actual robots. Every agent emits only one message during the execution of this algorithm – at most n for the entire system.



1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8	0	0	0	0	0	0	0
	1/4		0	0	0	0	0	0	1/4	1/4	1/4	0	0	0
		1/4	0	0	0	0	0	1/4	0	0	1/4	1/4	0	0
			1/4	0	0	0	0	1/4	0	0	0	0	1/4	1/4
				1/4	0	0	0	0	1/4	0	1/4	0	1/4	0
					1/4	0	0	0	1/4	0	0	1/4	0	1/4
						1/4	0	0	0	1/4	1/4	0	0	1/4
							1/4	0	0	1/4	0	1/4	1/4	0
								1/2	0	0	0	0	0	1/2
									1/2	0	0	0	0	1/2
										1/2	0	0	0	1/2
											1/2	0	0	1/2
												1/2	0	1/2
													1/2	1/2
														1

Figure 3: I-state transition graph and corresponding transitional probability matrix for the case of $h = 4$ wireless, $s = 3$ location IDs. Each graph node contains an $h \times s$ state matrix and each edge is labeled with the number of observations which trigger the given transition.

Logarithmic Algorithm

The Logarithmic algorithm runs for $\log_2(n)$ time steps and generates at most $\frac{1}{2}n\log_2(n)$ messages. This is due to the fact that at most half of all robots will emit a message pair at each time step, since only half of all possible unique IDs will have a 1 in their i^{th} bit.

Probabilistic Algorithm

The one-to-one association problem for a neighborhood of size $h \times s$ has a lower bound of $ceil(\log_2(s))$ time steps. The best case scenario for the association problem occurs when each observation made by the listener splits all current partitions of the i-state bi-graph S^i evenly.

Due to the probabilistic nature of this algorithm, its overall analysis is not straightforward and a closed-form solution for its expected running time has not been discovered.

Mathews et al. (2010) present a probabilistic algorithm which can be modeled as a branching process and show an approximate upper bound for the mean of the expected time. Robots participating in the algorithm solving this model "die-off". The authors emphasize the difficulty in finding a closed form solution for the probability density function of the process.

We were unable to apply this model to the more general system we present here. The difficulty arises from successfully identifying elements of our model which can be regarded as "individuals" in a population which can be discarded (produce zero offspring).

Another approach is to use the **transition probability matrix** corresponding to the directed i-state transition graph. The graph has the necessary properties of an absorbing Markov chain (Grinstead and Snell, 1997) — it has an absorbing state (the final state) and this state can be reached from any other state in the graph (transient states). See Fig. 3 for an example of the transition probability matrix. The expected number of steps from any state to reaching the final state can be calculated using the following formula:

$$\mathbf{t} = (I - Q)^{-1}\mathbf{1} \qquad (1)$$

Here Q is a $t \times t$ matrix containing all transient to transient transition probabilities, $\mathbf{t}$ denotes a vector of size t, I is the identity matrix, and $\mathbf{1}$ is a column vector of all 1's. The i^{th} element of $\mathbf{t}$ gives the average expected number of steps to absorption if the process started from the i^{th} transient state. Thus, Eq. (1) can be used to calculate the expected running time to complete the one-to-one association problem for a given neighborhood.

This approach, however, is prohibitively expensive for all but the smallest values of h. As was noted earlier, the number of i-states (t) for a given neighborhood is given by Bell's number: B_h. Berend and Tassa (2010) give an upper bound of Bell's number which is $O(h!)$ and $\Omega(2^h)$. A simple application of all possible observations to all possible i-states can generate the matrix in $O(2^h B_h)$ time. The matrix inversion needed in Eq. (1) will take $O({B_h}^3)$. The absorbing Markov chain method, therefore, runs in $O(h!^3)$.

A third approach is to simulate the running of the algorithm and gather statistical data. This was done by simulating activations of a neighborhood and running the Probabilistic algorithm until a one-to-one association was made or a threshold was reached. Each neighborhood was run between $50,000$ and $100,000$ times. We simulated all neighborhood size combinations of $h \in [2, 85]$ and $s \in [1, h-1]$. To validate the date from the simulation we also calculated the precise expected value for all combinations of $h \in [2, 8]$ and $s \in [1, h-1]$ using the absorbing Markov chain approach. The absorbing Markov chain method results matched the simulation results up to the precision we used during the calculations (0.0001 time steps).

The results from the simulation trials of neighborhoods up to size 40 can be seen in Fig. 4, the remaining trial data is omitted to keep the figure more readable.

The results from the simulation can be informally approximated (further work is needed to get a good fit for the data) by the following two functions which give the expected time steps to complete the one-to-one association problem:

1. $2 \log_2(h) \log_2(s+1)$
2. $\operatorname{arsinh}(hs)$

Using the first function, we can approximate the number of expected messages a given robot will emit: $0.5 \times 2 \log_2(h) \log_2(s+1)$. The h and s would be based on the longest lasting neighborhood the robot participates in and assumes that the robots stop transmitting messages once all their neighbors inform them that they have completed the one-to-one association. For a dynamic system, this assumption would not be true and the robot may continue transmitting messages after the initial association has been completed.

Algorithm Comparison

The two deterministic algorithms have a performance which is dependent on the global size of the system. They also provide an exact execution time for the one-to-one association problem. The Naïve algorithm gives us the slowest performance but uses the least amount of messages. The Logarithmic algorithm is faster than the Naïve one but requires an increase in the number of messages used. The two deterministic algorithms also require very minimal use of the wireless channel (only used during the first system wide transmission, which is not strictly required for the Logarithmic algorithm).

The probabilistic algorithm does not depend on the global size of the system. Every robot can quickly achieve the one-to-one association for its neighborhood. This algorithm has the disadvantage that it has no guaranteed time expectancy. The overall performance of the system is dependent directly on the types of neighborhoods that compose it so an analysis of the robot distribution may be required before choosing this algorithm.

Relaxing Assumptions and Extending the Formalism

In this section we discuss relaxing some of the assumptions introduced in the course of the paper. We show that after

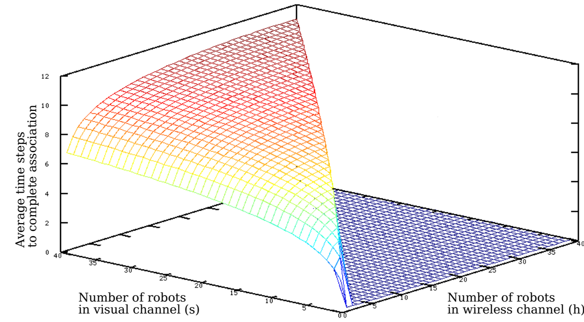


Figure 4: This is a plot of the average number of time steps to complete the association problem for all $h \times s$ neighborhoods for which $h \in [2, 40]$ and $s \in [1, h-1]$. Every neighborhood was simulated between 50,000 and 100,000 times to obtain the results

relaxing these assumptions the formalism still applies and the algorithms can still be used with minor modifications.

We have assumed that robots operate in an environment with no noise, communication errors or failures. This assumption guarantees that when a robot sends a message it is always received. This results in edges in the i-state (possible associations) being either 0 or 1. If we allowed noise, communication errors and failures, an observation may carry false positives or false negatives. The formalism and probabilistic algorithm can still handle this case. However, instead of a 0 or 1 stored as a possible association between two IDs, an appropriate statistical measure is computed and stored for each possible association. This makes the bipartite graph representing an i-state a weighted one. The finite information space interpretation presented in this paper is not directly applicable to this case.

The channel inclusion property imposes restrictions on how the coverages of the communication channels are related. The assumption in this paper guarantees that a robot's continued observation will result in an i-state in which all robots in visual (channel 2) range will be matched to robots in wireless (channel 1) range. This restricts the possible final i-states. If the assumption is relaxed fully, the resulting final i-state may be an empty i-state (a fully disconnected bipartite graph) indicating that the sets of robots in each channel are completely disjoint. The formalism will still correctly represent the system. Perhaps more surprisingly, all three algorithms correctly handle this situation with no modification other than a modified stopping condition.

We specify that there are two communication channels. The addition of more channels can be beneficial for the Probabilistic algorithm, as it can partition the i-state bipartite graph faster. This, however, would require the i-state's dimensions to grow as the number of channels. In this work we have not pursued any analysis on the impact on performance that the addition of channels will have.

The visual channel is assumed to be a binary channel. If the visual channel can distinguish between several colors, as done in (Mathews et al., 2010), the effective bit-rate of the visual channel increases as the color can be used to pass additional information. For example, the message pair can contain a colored bit in the visual channel and a color label in the broadcast channel. This will result in a faster partitioning of the i-state bipartite graph.

Property 1 allows robots communicating through a communication channel to share IDs. For example, a system in which robots are guaranteed to not move in a way that will bring two duplicate IDs in the same neighborhood can operate with no conflicts. A system where positions can change arbitrarily will require the stricter unique ID requirement. If an outside reference, such as geographical data or some environmental property, is used to assign IDs a system can have some reuse of IDs and a guarantee that a conflict will not occur.

Conclusion

In this paper we consider an association problem between IDs in two independent communication channels in a multi-robot system. The first channel is a radio channel which carries indexical knowledge about the robots while the second channel is a situated one and carries objective knowledge about the robots. Robots in the system need to make a one-to-one association between the two types of IDs as a preprocessing step to solving system tasks.

We provide a detailed formal description of the system and examine its properties from an information filtering perspective. We also show that this problem can be solved quickly and without the need for global localization knowledge and global communication by describing and analyzing two deterministic and one probabilistic algorithms.

The analysis of the probabilistic algorithm showed us that a closed-form solution for the expected performance of the algorithm based on the absorbing Markov chain is prohibitively expensive and that the applicability of another — based on branching processes — is uncertain but needs to be considered in more depth. We provide simulation results and give two functions which provide an informal approximation to the algorithm's performance. We compare the three algorithms and show that each has its advantages and drawbacks, relating to speed of execution, messages sent and communication channels used.

Lastly, we examined the ability of the formalism and algorithms we described to handle more realistic scenarios by relaxing some simplifying assumptions made during the creation of the formalism and analysis of the algorithms.

One future extension of this work is to examine in more detail the applicability of branching processes to the system model defined in this paper. Another extension is to analyze the performance of the algorithms after some assumptions, such as no communication failures, have been relaxed. A third extension is finding a better fit to the simulation data which can be used to predict the behavior of larger multi-robot systems operating under the probabilistic algorithm.

References

Agre, P. E. and Chapman, D. (1987). Pengi: An implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 1*, AAAI'87, pages 268–272. AAAI Press.

Berend, D. and Tassa, T. (2010). Improved bounds on Bell numbers and on moments of sums of random variables. *Probability and Mathematical Statistics*, 30(2):185–205.

Cassinis, R. and Tampalini, F. (2007). AMIRoLoS an active marker internet-based robot localization system. *Robotics and Autonomous Systems*, 55(4):306 – 315.

Dieudonné, Y., Dolev, S., Petit, F., and Segal, M. (2009). Deaf, dumb, and chatting asynchronous robots. In *Proceedings of the 13th International Conference on Principles of Distributed Systems*, OPODIS '09, pages 71–85, Berlin, Heidelberg. Springer-Verlag.

Fox, D., Burgard, W., Kruppa, H., and Thrun, S. (2000). A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344.

Franchi, A., Oriolo, G., and Stegagno, P. (2009). Mutual localization in a multi-robot system with anonymous relative position measures. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3974–3980.

Garrido-Jurado, S., Muñoz Salinas, R., Madrid-Cuevas, F. J., and Marín-Jiménez, M. J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recogn.*, 47(6):2280–2292.

Grinstead, C. M. and Snell, J. L. (1997). *Introduction to Probability*, chapter 11: Markov Chains. American Mathematical Society.

Haccou, P., Jagers, P., and Vatutin, V. (2007). *Branching Processes: Variation, Growth, and Extinction of Populations*. Cambridge Studies in Adaptive Dynamics.

Howard, A., Parker, L. E., and Sukhatme, G. S. (2004). The sdr experience: Experiments with a large-scale heterogenous mobile robot team. In *9th International Symposium on Experimental Robotics 2004*, Singapore.

LaValle, S. M. (2009). Tutorial: Filtering and planning in information space. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IROS'09, pages 1–1, Piscataway, NJ, USA. IEEE Press.

Lespérance, Y. and Levesque, H. J. (1994). Indexical knowledge and robot action - a logical account. *Artificial Intelligence*, 73:69–115.

Mathews, N., Christensen, A. L., Ferrante, E., O'Grady, R., and Dorigo, M. (2010). Establishing spatially targeted communication in a heterogeneous robot swarm. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, pages 939–946, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Støy, K. (2001). Using situated communication in distributed autonomous mobile robotics. In *In Proceedings of the 7th Scandinavian conf. on Artificial Intelligence*.