

# Distributed Heterogeneous Robot-Human Teams

Robotics Track

S M Al Mahi

Department of Computer Science,  
Oklahoma State University  
Stillwater, Oklahoma  
smahi@okstate.edu

Kyungho Nam

Department of Computer Science,  
Oklahoma State University  
Stillwater, Oklahoma  
kyungho.nam@okstate.edu

Christopher Crick

Department of Computer Science,  
Oklahoma State University  
Stillwater, Oklahoma  
chriscrick@cs.okstate.edu

## ABSTRACT

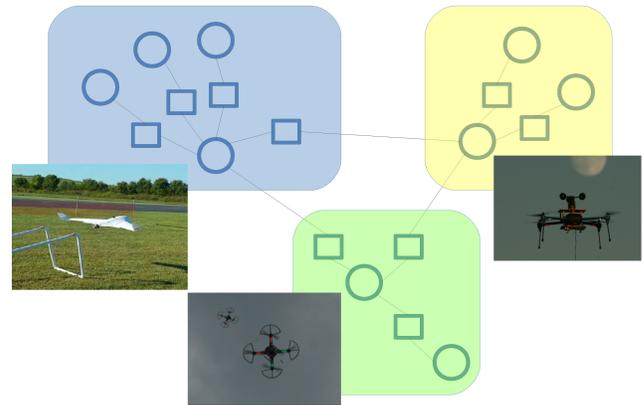
We introduce a novel, scalable, distributed decision-making algorithm using factor graphs and the sum product algorithm to control the coordination of a heterogeneous multi-robot team in exploration tasks. In addition, our algorithm supports seamless participation of human operators at arbitrary levels of interaction. We present experimental results performed using both simulated and actual teams of unmanned aerial systems (UAS). Our experiments demonstrate effective exploration while facilitating human participation with the team. At the same time, we show how robots with differing capabilities coordinate their behaviors effectively to leverage each other's individual strengths, without having to explicitly account for every possible joint behavior during system design. We demonstrate our algorithm's suitability for tasks such as weather data collection using a heterogeneous robot team consisting of fixed- and rotary-wing UAS. In particular, during 60 flight hours of real-world experiments collecting weather data, we show that robots using our algorithm were about seven times more efficient at exploring their environment than similar systems which flew preplanned flight profiles. One of our primary contributions is to demonstrate coordinated autonomous control and decision-making among robots operating in very different flight regimes.

### ACM Reference Format:

S M Al Mahi, Kyungho Nam, and Christopher Crick. 2019. Distributed Heterogeneous Robot-Human Teams. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13-17, 2019*, IFAAMAS, 9 pages.

## 1 INTRODUCTION

In many applications, it is advantageous to have a heterogeneous group of agents cooperating in their execution of tasks and their search for solutions. For example, a search and rescue robot team might incorporate unmanned aerial systems (UAS) to survey and map the affected area, small ground robots to infiltrate and explore rubble and pipes, large ground robots with manipulators to clear heavy objects, and friendly-looking social or medical robots to locate and make contact with the injured. Even similar robots might have very different kinds of capabilities and sensors, such as mapping sensors, hazardous chemical sensors, or sonic sensors for voids and places where people could be trapped. A heterogeneous robot team like the one described above is often difficult and expensive to engineer correctly, especially if the agents possess very different



**Figure 1: A team of heterogeneous UAS with meteorological sensors share a single factor graph representation of their shared intentions.**

physical characteristics and abilities. In this paper, we are motivated primarily by the problem of using UAS to measure various aspects of weather systems. A large number of different kinds of UAS with different capabilities and sensors may be employed. For example, the system may incorporate large, fixed-wing UAS with a variety of sensors, smaller and less expensive rotary-wing UAS, disposable single-use sensors deployed with parachutes, and many airframe and sensor variations within these categories. Additionally, as projects like this develop, new varieties of agents may be added after the system is already in place.

The conventional way to approach this kind of problem would be to individually tailor the software for each type of agent to produce cooperation and other desired behavior. For example, [3] in search and rescue (SAR) tasks, the logic of these two types of robot tasks are designed separately. This approach is often adequate when the different types of agent are not intended to change across missions and the desired collaborations are simple. However, as the number of agent types increases and the problems become more complex, this approach rapidly becomes unsustainable. The complexity of the entire system grows quadratically with respect to the number of agent types, potentially resulting in an unmanageable error-riddled code base and vastly increased cost and development time. Furthermore, such an approach becomes completely unmanageable when new varieties of agents are added after the system is deployed, since it requires significant modification to the logic of all other

agent types every time a new variety is introduced. Our approach places these heterogeneous sensor and actuation modalities within a uniform multidimensional belief manifold represented by a factor graph shared among all of the agents in the system. The dimensions of the manifold can (and in many use cases, probably will) represent positions in physical space, but can also encode agents' joint intentions over any arbitrary space of action potentials. Actions to take with respect to other agents, obstacles, unexplored areas and points or gradients of interest can all be encoded in these intentional beliefs. The agents communicate in decentralized and asynchronous fashion, using loopy belief propagation to update the team's joint intentional state. Consensus beliefs are then acted upon by each agent in the local area of the manifold using simple gradient descent.

Many other similar tasks like SAR might require coordination of heterogeneous multi-robot teams with arbitrary human intervention. The task we have particularly investigated is surveying atmospheric data in the lower altitude (under 1000 feet) boundary layer[10]. Although satellite images, weather towers and balloons are used to collect weather data, they are not able to provide coverage in this important area of the atmosphere, even though sudden developments within the boundary layer can significantly contribute to severe weather formation. UAS have been used by meteorologists to collect data in the boundary layer, but they are manually controlled by human operators and usually follow a predefined simple mission plan, such as vertical profiles or circular orbits. Such an approach has drawbacks in efficiently deploying multiple robots and in fielding sufficient trained human operators. Proper coverage requires a multi-robot deployment, and spatiotemporal data can change dynamically, requiring multiple sensors spread out in space to properly sample time-varying data. Assigning human operators to each UAS is not practical, as robot participants must coordinate closely with one another and make data-dependent decisions across the entire robot team in real time. We have addressed all of these problems using our factor graph algorithm. The application of our proposed algorithm is not limited to these particular tasks, but also can be generalized to many other tasks requiring heterogeneous multi-robot teams cooperating with human operators.

In many human-robot team tasks, the human operator cannot inform the robots of his or her intentions efficiently because of the different facets of the task exposed to the human operator and the robots. Similarly, it is often challenging to create an interface through which robots can efficiently convey their knowledge and intentions to their human operators, especially when the participant robots are heterogeneous in nature. This paper provides a theoretically neat, practically robust, and generally efficient model for heterogeneous, scalable and dynamic human-robot teams.

## 2 RELATED WORK

A great deal of research has been conducted on multi-robot navigation, exploration and surveillance in different applications. Several recent works [18, 19] aimed at monitoring lower altitude atmospheric variables and sampling weather data using UAS and ground robots. However, very little work has been done to effectively coordinate multiple collaborating robots for such tasks, which demand

fault tolerance, scalability, autonomous decision-making and human incorporation together. Our research is motivated by collecting weather data in the low-altitude (under 1000 feet) atmospheric boundary layer in coordinated fashion. This requires a heterogeneous team of robots equipped with various sensors. Because of the large permutation of capabilities and constraints of these robots, human operators are subject to intense cognitive load while operating them. Teams of robots must often make autonomous decisions, even when attempting to satisfy conflicting mission goals. In this paper, we have proposed a loopy belief propagation [7] algorithm within a shared factor graph model. In our decentralized approach, no particular node in the factor graph is essential as long as the network maintains redundant communication pathways, and every robot continually updates its own intentional model with messages it happens to receive from robots within communication range. Thus our approach is fault tolerant as well as scalable. The computation required in loopy belief propagation can be distributed among different agents and each robot's computational requirements are much simpler compared to other methods[25] based on techniques such as Markov decision processes. Thus there are no technical barriers to adding arbitrary numbers of additional robots to the team. It is scalable in the spatial sense as well; our live experiments have been conducted in volumes as large as several kilometers across.

Using belief networks as coordination tools for multiple robots has been proposed in the past[5], usually in the context of Markov random fields [4, 14, 24]. However, our factor graph representation provides several advantages. First of all, the functions defined within the factor graph are often very simple to engineer. For example, collision avoidance is one of the commonly desired behaviors in a multi-robot exploration task. Instead of explicitly designing collision avoidance mechanisms, such as in [8, 22], we merely design a simple factor graph function which reacts to obstacle positions. Using our approach, complex team behaviors can be specified, tested and changed quickly. In addition, our formulation explicitly allows for seamlessly injecting human directives and advice into the robots' shared intentional framework, as well as additional expected behaviors, at arbitrary levels of specificity and timeliness. Moreover, many variants of MDPs, notably Partial Observable MDPs (POMDPs) are known to be intractable in larger domains.

In our approach, the robots can make autonomous decisions by following the gradient in the joint belief over the space. Gradient-based multi-agent navigation has been studied in other works[9, 13]. However, those studies focused on designing specific goals for a particular task, facilitating expansion of the behaviors. We have extended the idea by designing joint beliefs which can be devised to achieve any expected behavior from the robot team: following human intentions, for example. Additional complex behaviors can easily be introduced by designing new functions as factors in the factor graph.

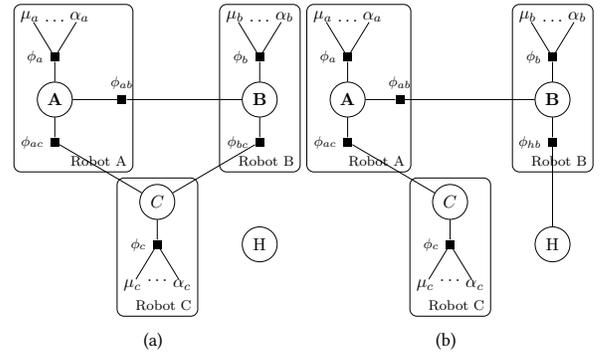
The problems we consider involve heterogeneous communicative robots [21]. Such robots can be seamlessly incorporated into teams which can then form joint plans and task allocations around each participant's capabilities. This resource allocation problem has been studied extensively [1, 2, 23] from the underlying network topological perspective. Distributed robot teams are commonly considered to be nodes in a network, with connections among the

various robots represented as edges. These edges represent communication links between robots or express other relationships of interest. This work demonstrates the graph-theoretic properties necessary in task allocation and team configuration of heterogeneous multi-robot teams. Our distributed probabilistic graphical model easily incorporates heterogeneity because a robot only adds sensor data that it is capable of gathering and only forms intentions over actions it is physically able to take, while continuing to update the factor graph and propagate the information provided by other agents with different capabilities.

For the simulation of multi-agent systems, ROS- Gazebo[11, 20], an open source simulation system, has been used by many academic researchers. It can be used for simulating motion planning in indoor or outdoor environments for teams of ground robots and UAVs [6]. However, it does not have built-in capability for simulating real-world outdoor environments, with varying winds, visibility, air density and turbulence. Moreover, one of our research focuses is to achieve a distributed control system that can be used even in harsh environments such as during severe storms. Simulating many of these environmental phenomena requires a huge amount of weather simulation work if it must be developed from scratch. A few other common problems with other simulators, with respect to our research interest, are extendability and availability. For example, AgentFly[16] a popular multi-agent flight simulator, facilitates control and planning in constrained environments. However, its control system is not customizable and it is not free. Many other commercially available simulators are expensive and out of reach of research community in general. We have used Flight Gear, which is an open source, freely available flight simulation platform mostly for fixed wing aircraft and helicopters. We have extended and customized it with various UAS flight models. Importantly, it also provides a customizable weather and visibility engine, which we have used to simulate specific boundary layer atmospheric phenomena, as well as 3-dimensional gas plumes from methane or carbon dioxide releases – phenomena which we are also able to generate and test in real-world applications. We report results in this paper from both our simulated system and a very extensive set of real-world experiments.

### 3 PROBLEM FORMULATION

Consider robots A, B, C in an autonomous multi-robot team. Figure 2(a) shows an example of a factor graph. In this graph A, B, C and H represent four variable nodes of the factor graph. Among the variable nodes A, B, C represent the intentions of the respective robots, while H represents intentions provided by a human operator. A robot builds its own belief from its sets of sensors. For example, robot A builds its belief about the world using its set of sensors  $\mu_a$  and actuators  $\alpha_a$  and so on. These sets are not necessarily the same for all the robots in team, and thus the model naturally includes heterogeneous robots. Using  $\mu_a, \mu_b, \mu_c$  robots A, B, C can build their own beliefs about the state of the world and can form their own intentions. This can be done using factor functions, i.e.  $\phi_a, \phi_b, \phi_c$  respectively. In Figure 2(a), all the robots are connected to each other in the sense that they can communicate with each other and incorporate each others' intention representation in calculating a joint distribution of belief.



**Figure 2: A factor graph model of our proposed algorithm with three robots and one human operator. Robots A, B and C compute intentional representations (in circle nodes), maintain sensor and actuator observations ( $\mu$  and  $\alpha$ ), and apply  $\phi$  functions to messages passed around the network. A human participant is labeled H. (a) A factor graph where all the robots are connected to each other. No human intentions currently being provided. (b) A is connected to B and C. No connection between B and C. The human operator can communicate additional intentions to B, which are shared throughout the robot team via loopy propagation.**

For example, the intention of robot B is incorporated into A's belief using factor function node  $\phi_{ab}$ . In a joint exploration task for robots A, B and C, the collective objective might be to explore the entire space as quickly as possible while maintaining a collision-avoidance distance from each other. In such a scenario  $\phi_{ab}, \phi_{bc}, \phi_{ca}$  can be designed as functions that make a space which has been visited by a particular robot less interesting for the other robots. For example, such a function could compute a time-decaying penalty associated with the robots' various reported location observations, while collision avoidance could be represented as a much stronger penalty function computed from a robot's current position and velocity. Similarly, other types of goals can be achieved using differently-devised  $\phi$  functions. We will demonstrate this in our experiments. Robots can communicate their intention by passing messages to other neighboring robots, and they, in turn, pass that information along to other robots in the team.

In general, the joint belief  $g$  can be calculated using Equation 1,

$$\begin{aligned} g(x_1, \dots, x_n) &= \prod_{j \in J} f_j(X_j) \\ &= \frac{1}{Z} \prod_{ij} \phi_{ij}(x_i, x_j) \prod_{ij} \phi_i(x_i) \end{aligned} \quad (1)$$

where  $f$  is a generic function of the set of all the variables  $x \in X$ . By definition, a factor graph[12] is a bipartite graph of variables and factor functions. The computation of Eq. 1 can be performed using a loopy belief propagation (LBP)[15] message passing algorithm on a factor graph. Here  $\phi_i(x_i)$  denotes robot  $i$ 's belief about variable  $x_i$  in the world from its sensory information.  $\phi_{ij}(x_i, x_j)$  is the belief robot  $i$  forms from information received from robot  $j$ . All the  $\phi$  functions are the factor functions in the factor graph. The

loopy belief propagation algorithm on a factor graph is shown as Algorithm 1.

---

**Algorithm 1: Loopy Belief Propagation LBP**


---

```

Function Main( $i$ ):
  /* at  $i^{th}$  robot */
  repeat
    AsynchronousUpdate()
     $\pi_i^t = \nabla g(x_1, \dots, x_n)$  /*  $t$  signifies time */
    ExecutePolicy( $\pi$ )
  until

```

```

Function AsynchronousUpdate( $i, j$ ):
  Update belief using Eq.2;
  Update intention using Eq.3 and broadcast
return

```

---

$$\mu^{a_i \rightarrow a_j}(x_j) = \frac{1}{Z} \sum_{x'_i: x'_v = x_v} \phi(x'_i) \prod_{V \in N_i \setminus a_v} \mu^{a_v \rightarrow a_i}(x'_v) \quad (2)$$

Equation 2 defines the message  $\mu$  passed from a variable to a factor, which consists of the normalized product of all of the messages received from the variable's neighboring factors, except for the recipient factor. Set  $N_v$  denotes the neighboring participating robots in the team for a robot  $a_v$ .

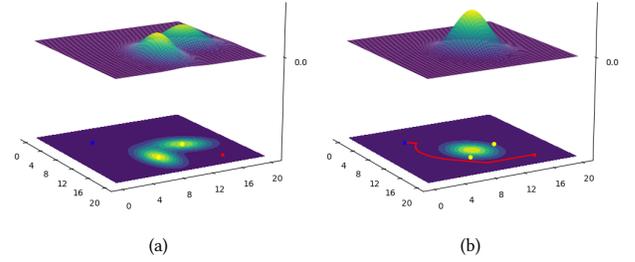
$$\mu^{a_i}(x_i) = \frac{1}{Z} \phi_i(x_i) \prod_{k \in N_i} \mu^{k \rightarrow a_i}(x_i) \quad (3)$$

Equation 3 shows the message  $\mu$  passed from a factor to a variable, which is the factor function applied to the messages from all other connected variable nodes, marginalized over all of the variables except the recipient's. These messages are passed asynchronously through links that are formed and dropped as the topology of the robot deployment changes, using loopy belief propagation. At a certain time  $t$  a particular robot  $a_i$  can run an optimization algorithm locally on the joint distribution of intention using its policy  $\pi_i$ .

$$\pi_i^t = \nabla g(x_1, \dots, x_n) \quad (4)$$

Low-level controllers such as PIDs can achieve the goal produced by Eq. 4. We used a gradient descent algorithm to calculate  $\pi_i^t$ . Other methods can be used as well.

A human operator can be imagined as another factor variable node in the overall graph, although they are not responsible for performing any computation (note that the  $\phi_{ha}$  function applied to the human input in Figure 2(b) rests within the computational responsibilities of Robot B). Incorporation of human operators in the distributed factor graph coordination is one of the major contributions of this paper. Our approach allows a human operator to exert an arbitrary amount of control over all of the agents that are indirectly or directly connected to the operator. If no human input is available (for example, if the human operator is task-saturated or does not have a connection to the agent), then the agent and the



**Figure 3: (a) Two robots' observations regarding the same obstacle are represented as Gaussian distributions over navigation space. (b) After integrating the beliefs according to its  $\phi$  function, one robot (blue star) executes a policy which leads to a goal (red star).**

entire system function autonomously according to the robots' own sensor data and communicated beliefs.

When more guidance is available, the system will continue to propagate messages in identical fashion, but human input will be seamlessly integrated into that process in the form of one or more additional intentions mediated by an additional factor function. Furthermore, if the operator only has an indirect connection to an agent through other agents, these imposed beliefs will still reach the appropriate agent through loopy belief propagation within the whole network. The fusion of the distributed communications architecture, the belief-based information processing, and the optional human interaction allows us to create a general purpose heterogeneous architecture that accommodates smooth changes in robots, hardware, and human input.

One of the known limitations of loopy propagation within factor graphs is that the beliefs occasionally fail to converge in certain cases. This problem is rare, and we went to considerable effort to evaluate its effect on our particular application. We simulated twenty autonomous UAS systems communicating their beliefs about their and each others' observed positions, with many different belief parameterizations. Each robot communicated with all the others, forming the loopyest possible clique. Even so, in hundreds of trials, we were never able to induce the graph to fail to converge. Convergence was always achieved within eighty message iterations.

In our real-world experiments, we have so far involved only as many as five robots, which makes the problem even less likely. In addition, the robots are functioning in real time, with changing conditions, positions and measurements, so if they were to find themselves in a rare non-convergent state, they would quickly emerge from it before it had a chance to produce a substantial performance or safety impact. We can calculate a theoretical upper bound for the probability that the propagation algorithm produces a problematic result. If we assume that nonconvergence induces the absolute worst possible policy selection (itself hugely unlikely), then a collision could happen with probability  $p(NC)^n$ , where  $p(NC)$  is the probability of the network entering a non-convergent state and  $n$  is the number of time steps it takes to steer straight for an obstacle (as opposed to away, as the actual computed policy would

indicate). Likewise, for a survey problem, the worst policy choice would mean moving in the least interesting direction instead of the most, which could increase the time taken by  $2np(NC)$ , where in this case  $n$  is the number of time steps to conduct the survey. Since our simulations indicate that  $p(NC)$  is a number extremely close to 0, these negative effects are negligible.

## 4 EXPERIMENTS

We have conducted several experiments using simulated and real robots. In this section we describe three tasks involving exploration of the spatial environment, starting with simple intentions such as exploration and collision avoidance, and demonstrate how additional complex intentions can be introduced easily using our approach.

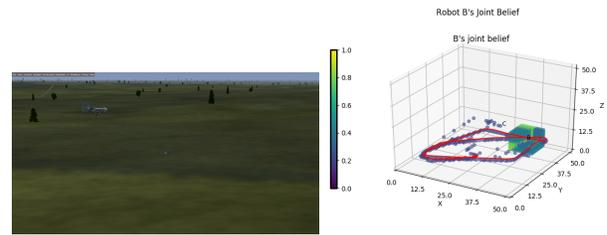
### 4.1 Exploration Within Boundary Avoiding Collision

In the first experiment we implement the basic functionality of our proposed algorithm for a team of simulated heterogeneous UAVs, conceived as a collection of ROS[17] nodes which pass factor graph messages between each other over ROS topics. It has been implemented in a space discretized system. The messages passed from one robot to other include the UAV's current position, a history of positions it has already visited, and sensory information such as temperature, humidity and air pressure. For this first experiment, the robots do not make decisions based on the sensory information, but they will do so later in the paper. The messages passed by each robot were also timestamped, which can be used by  $\phi$  functions which weight information by recency. The messages passed were lists of parameters packed in ROS's message format. For all the experiments, the messages included at least the robots' position, orientation, the identities of neighbors and a limited history of previously broadcast parameters. We have also run experiments using non-parametric methods where the intentions over the whole navigation space were shared as a probability mass function. However, we have not used non-parametric methods in the experiments discussed in this paper. In the experiments discussed later in this paper, the messages also included sensor readings like temperature, humidity,  $CO_2$  density and so on. For fixed-wing aircraft, roll, pitch, and yaw angles along with GPS position and other aerodynamic parameters were passed, which were incorporated into an agent's belief calculations.

We have extended the open source flight simulator Flightgear<sup>1</sup> in order to develop a robotic simulation software suite which supports atmospheric physics phenomena such as turbulence, visibility, temperature, humidity, and the behavior of water vapor and gas plumes such as clouds, smoke, methane, and carbon dioxide. Figure 4 shows simulated UAVs which have an exploration task while avoiding collision and remaining within a constrained volumetric boundary – in this case,  $500m^2$ , between altitudes of 50 and 450 feet. Much larger scales are algorithmically tractable. The whole space was discretized into  $50 \times 50 \times 50$  voxels.

In this experiment, three basic  $\phi$  factor functions are used for each robot, i.e.  $\phi_u$  (unexplored),  $\phi_b$  (boundary) and  $\phi_{ac}$  (avoid collision).  $\phi_u$  is a local weight function applied to a space whenever

<sup>1</sup><http://home.flightgear.org/>



(a) Simulated fixed-wing UAV (denoted B) avoiding collision with quadrotor (denoted C)  
(b) Joint belief of fixed-wing UAV B. Red line shows path B followed. B only calculates local gradient over the joint distribution of belief.

**Figure 4: Heterogenous robots avoid each other while exploring the space.**

a robot visits, making it less interesting.  $\phi_b$  is a function which has high value near the boundary of the space and zeros everywhere else.  $\phi_{ac}$  is a Gaussian distribution with mean at a robot's current position and a standard deviation of 3.33 and 5 voxel units for quadrotor and fixed wing UAV respectively. The negligible probability mass beyond  $3\sigma$  is ignored. The parameters were chosen based on the various platforms' maximum airspeed and the size of the voxels, sufficient for belief updating at 10 Hz to allow the robots to take autonomous actions before a collision happens. A robot takes a normalized weighted sum of these functions derived from both its own sensors and the messages from its neighbors to build a joint intention over the space local to its current position. The weight  $w_{\phi} \in (0, 1)$  of each intention  $\phi$  depends on the priority of the task. For example, the unexplored function will have less probability mass in unvisited areas, attracting the robot to navigate there. However, if it is close to another robot's current position then the probability mass from collision avoidance will outweigh the attraction because the  $\phi$  functions associated with avoiding collisions and avoiding boundaries are accorded much higher weight than other behaviors not concerned with safety of flight. These weights also help humans to control the behavior of the the autonomous system. The robot descends the gradient of its joint belief. For all of our applications, boundary weight was set to 1. Weights for direct and indirect collision avoidance were set to 0.85 and .8 respectively. The direct collision avoidance parameter is used for a robot which is directly communicating its position to another robot, whereas the indirect parameter is used to weight the collision avoidance intention from other robots that are propagating their position through an intermediary node or robot.

Fig. 4 shows the fixed-wing UAV, denoted as B, avoiding collision with the quadrotor, C, despite the quadrotor being placed directly in the path of the fixed-wing UAV. It also depicts the joint intention that B builds using messages it received from its neighbor C. The scatter plot shows the distribution of intentions of the robots over the exploration space. The color-coded voxels signify the probability of a voxel being less interesting or worth visiting. The  $\phi$  functions and weights described for this experiment have very similar implications for the experiments we will discuss later.

**Table 1: Heterogenous  $CO_2$  plume mapping**

Location of Plume			Number of UAVs	Mapping Time (approx)	
X	Y	Z		Homogeneous Team	Heterogeneous Team
40	30	18	5	>20 min	20 sec.
7	28	8	5	3 min	120 sec.
26	34	34	5	>20 min	75 sec

### 4.2 Exploiting Heterogeneity

Using our factor graph distributed algorithm, we can exploit the heterogeneous configuration of our robot team. Teams of such robots can accomplish more complex tasks more quickly, and in distributed fashion. In this experiment, we demonstrate this using our simulator. The task is to locate, survey and map a  $CO_2$  plume within a given area. Our heterogeneous team consists of two similar fixed wing UAVs and three quadrotors with slightly different sensory capabilities. All of these simulated UAVs are equipped with GPS, temperature and humidity sensors, but only the fixed-wing aircraft and one quadrotor are equipped with  $CO_2$  sensors.

A fixed-wing aircraft is much faster than a quadrotor, but also far less maneuverable. A team of fixed-wing aircraft will quickly locate traces of  $CO_2$ , but they will not be able to carefully map its contours. On the other hand, while a maneuverable quadrotor is better equipped to perform the detailed survey, its slow speed makes the location of the plume difficult to find in the first place.

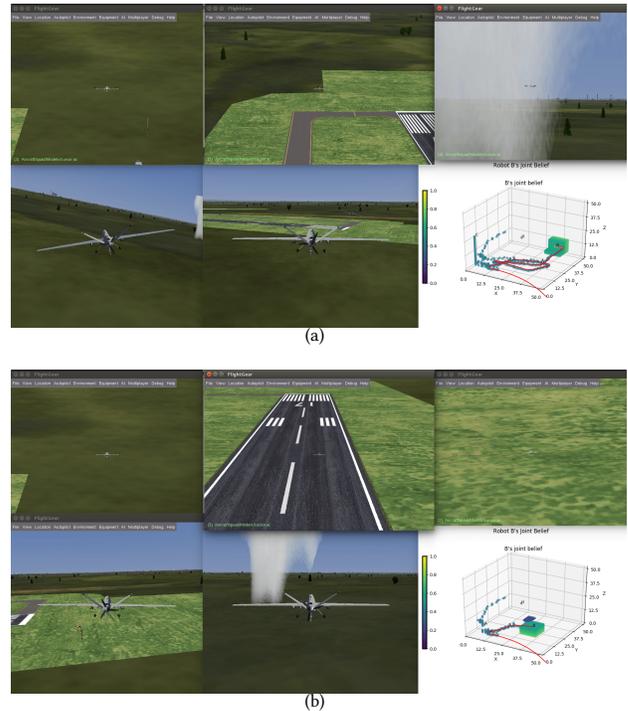
We have run the simulation in homogenous (all quadrotors) and heterogenous configurations several times with the  $CO_2$  plume situated in different locations. Table 1 compares the time taken to find and map the plume.

### 4.3 Experiment With Real Robots

We have used two 3DR Solo quadrotors<sup>2</sup> for conducting this experiment. Each is equipped with a DHT22<sup>3</sup> temperature and humidity sensor. To introduce heterogeneity, one of the sensors only reported temperature, while the other sensed humidity. The choice of these particular sensors were motivated by our interest in collecting data for atmospheric physics using UAS. However, other kinds of sensors could be supported using the same approach; our algorithm is software-based and is useful for many different hardware choices. Our experiment was conducted within a  $64 m^3$  cube of airspace, with a base altitude 5 meters above the ground for safety.

The  $\phi$  functions in place for this experiment were similar to those used on the simulator. A boundary  $\phi_b$  function places a very high value (meaning a strong avoidance intention) at the boundary of the cubic space. The boundary intention and the collision avoidance intention  $\phi_{ac}$  are both weighted very strongly. The collision avoidance intention has been designed again as a Gaussian probability mass function having the mean at the current location of a particular robot and a variance depending on relative robot speeds. The unexplored intention  $\phi_u$  assigns higher values to locations that

<sup>2</sup><https://3dr.com/solo-drone/>  
<sup>3</sup><https://www.adafruit.com/product/385>



**Figure 5: Snapshot of heterogeneous team exploration. For visibility we have hidden the joint intention of all the robots except for B (a fixed-wing UAS). Quadrotors and fixed-wing aircraft coordinate together to explore, and are each captured in different picture elements. The lower-right element is the joint belief of B. (a) UAS B is moving towards the  $CO_2$  plume. UAS A still exploring far from the plume. (b) UAS B has already passed through the plume and communicated intention information which can be exploited by UAS which are interested in exploring areas with high  $CO_2$  density. UAS A is such a robot so it moves towards the plume. B’s joint belief in the lower-right subfigure also shows the trail of A’s path.**

have been visited and are thus less interesting. The above  $\phi$  functions are very similar to the functions we have used in simulation. As this experiment incorporates temperature and humidity sensors, we provide two more intentions, namely,  $\phi_t$  and  $\phi_h$ , much as we did previously. However, we do not know the temperature and humidity over the whole space. We only have the measurements at the current location of the robots and the places they have been before, and those values may change over time. However, we can infer the temperature and humidity gradients over unvisited space using the data we have already collected. The robots will attempt to locate and explore areas of rapid change, as these inversion and boundary layer phenomena are most informative to a meteorologist.

We simulate these sharp temperature and humidity changes in our weather-aware simulation environment. Fig. 6(a-b) demonstrates that the robots are able to detect and map such sudden changes. Then, in Fig. 6(c-d), we show the same behavior in a real

world experiment. The robots are able to locate a temperature inversion at 45 meters above the ground.

We are also able to demonstrate seamless human intervention using our algorithm. In this particular instance, a human operator takes active control of one of the UAS, overriding the intentions developed by that robot. The robot, however, continues to communicate with the other team members, using the same loopy propagation framework. The other systems modify their intentions accordingly. Fig. 7 shows a human intentionally steering robot A toward its neighbor B. This induces robot B to evade, because of the influence of  $\phi_{ac}$ . The human’s intention is incorporated smoothly into the overall team behavior, without any explicit commands from the human to any other robot participant beyond the first.

#### 4.4 Experiment With *LOTS* of Real Robots

Our work contributed to the ISARRA Lower Atmospheric Process Studies at Elevation—a Remotely-piloted Aircraft Team Experiment<sup>4</sup> (LAPSE-RATE). Up to now, meteorologists and weather experts have used radar, balloon soundings and satellite data to model weather. However, as mentioned in the introduction, such sensor modalities cannot collect data in lower altitude environments effectively, and the atmospheric physics community does not have a good understanding of the boundary layer above the height of weather towers but below the safe operating envelopes of manned aircraft or the line of sight constraints of radar. We, along with more than 50 of our fellow researchers and colleagues from diverse research backgrounds and origins, participated in ISARRA LAPSE-RATE and collaboratively collected weather data using UAS. The experiment was conducted during a flight week campaign in the San Luis Valley in Alamosa, Colorado at several interesting sites from the perspective of atmospheric research. Collectively, 1200 flights were flown in this week by the participants, and our team contributed 215 flights and acquired 2.9 GB of sensor data. This is both the largest amount of data collected by UAS for weather measurement in a specific time period and geographic location, and involved more heterogeneous UAS platforms in the data collection than ever before.

One of our research focuses for this period revolved around testing the performance of our coordination algorithm and comparing its performance with the current approach, where meteorologists and roboticists collaborate to devise planned waypoint-based missions to map the weather environment, usually taking the form of fixed vertical profiles or transects between two geographic points at various altitudes. For this experiment, we performed 13 autonomous flights and 64 fixed, planned flights using several different quadrotor and fixed-wing UAS.

All of these autonomous, profile, transect and other flights were conducted in the San Luis Valley, west of Great Sand Dunes National Park, Colorado. The experiments were conducted following FAA guidelines, with special permission to fly missions involving swarms, night flight and high altitudes. All the flights were required to have a human FAA-licensed UAS pilot in charge of the flight. The autonomous flights were conducted in multiple locations over half-hectare areas of open farm fields or rangeland. The transect flights followed flight paths of approximately 1 KM in length. Although

**Table 2: Comparison Between Autonomous and Profile Flight**

Measure		Temperature	Humidity
Entropy	Autonomous	3.340759	2.997618
	Profile	3.424795	3.193233
Duration	Autonomous	05:17:11	05:06:03
	Profile	39:02:39	10:41:48
Info. Gain/sec	Autonomous	<b>0.0001755</b>	<b>0.0001632</b>
	Profile	0.0000244	0.0000256

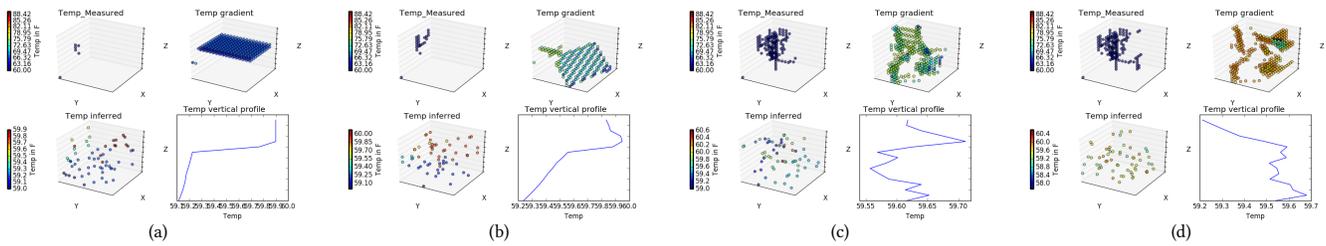
we have access to the data from all 1200 flights, these experiments were performed across sixty flight hours, a small fraction of the total.

The overarching goal of the LAPSE-RATE campaign participants, roboticists and meteorologists alike, was to discover interesting phenomena in weather data, such as sudden changes to temperature or humidity. This information allows atmospheric physicists to understand and model convective activity and other atmospheric behavior. We hypothesized that our algorithm, which is specifically intended to coordinate multi-robot teams in real time in response to heterogeneous sensor data, would be much more efficient at detecting and mapping these atmospheric phenomena, compared to UAS following fixed, preplanned profiles. We quantify the quality of the data collected by computing the entropy of the data distribution, and turn that into information gain per unit time. For this experiment we have used factor functions  $\phi_t$  (temperature),  $\phi_{ac}$  (collision avoidance),  $\phi_b$  (boundary), as well as  $\phi_h$  (humidity), which behaves just like the temperature function but is tied to a different sensor. We collected the sensory data into a normalized histogram, which essentially captures the probability density function of the changes of sensory data in the weather over time.

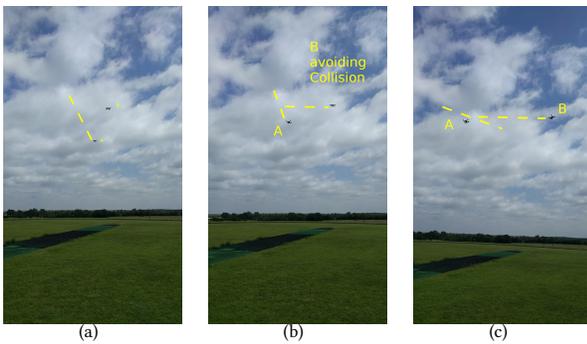
Figure 8 shows histograms computed for two representative flights. The overall information gain per flight is almost identical, and this is reflected in the similar shapes of the two histograms. However, as can be observed on the lower figure, the profile flight spent most of the second half of its trajectory in a very low information regime, where nothing particularly interesting was happening to the temperature. Thus the autonomously coordinated flight took half the flight time to obtain the same information.

Even a very poor deployment strategy for a meteorological sensor will continually produce information as it measures its environment over time. A better strategy will produce higher information gain over a specific amount of time. Thus we calculated the information gain density simply by dividing the entropy by the duration of the flight by seconds. This quantity tells us how well a particular flight worked in gathering interesting information about the weather quantitatively. Table 2 shows a comparison between our autonomous flight tests and pre-planned waypoint based profile flights. Autonomous flights following our algorithm have higher information gain and spend lower flight times to collect that information. Thus, they have a higher information acquisition rate on average than fixed profile flights. On average our autonomous

<sup>4</sup><https://isarra.colorado.edu/flight-week>



**Figure 6:** In each subfigure, upper left is measured temperature, upper right is the inferred temperature gradient, lower left is randomly sampled temperature predictions drawn from the inferred gradient, and lower right is a temperature vs altitude plot. a-b and c-d are temperature profiles in simulation and the real world respectively. In each case, the first figure is early in the exploration process, and the second is after additional exploration and mapping.



**Figure 7:** Experiment with two UAV robots A and B. (a) Human commands A to move toward B. (b) B moves to avoid collision with A. (c) A and B flying at safe distance again.

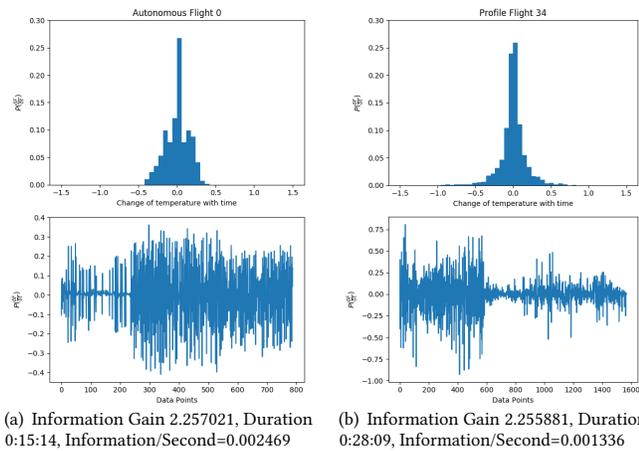
approach collected data around seven times more effectively for both temperature and humidity data. Because of the huge number of samples collected using both methods, our results are extremely statistically significant ( $p < 10^{-7}$ ).

### 5 CONCLUSION

In this paper, we have proposed a scheme for heterogeneous multi-agent control that uses factor graphs and loopy belief propagation to abstract intention away from the specifics of hardware capability and sensors, allowing a diverse collection of systems to be controlled with the same software and to interact effectively with each other. Additionally, human operators may insert themselves into the decision-making process to varying extents as desired. Our method enormously simplifies the logic and programming required to solve these kinds of problems. We have demonstrated the effectiveness of the approach in simple real-world scenarios and more complex simulated ones. At present, we have equipped actual UAV robots with real meteorological sensors and have demonstrated the efficacy of this approach in large real-world deployments, improving our understanding of near-surface weather phenomena and our ability to monitor and predict severe weather.

The contribution of this paper is the novel algorithm for distributed heterogeneous control of robots with humans in the loop, and a very large-scale experiment which confirms its applicability, performance and robustness. We also illustrate theoretical evidence of its performance and developed a simulated environment which makes it possible accurately to generate real-world weather phenomena for multi-robot UAS testing. Our work facilitates human interaction with heterogeneous multi-robot teams. We have run an enormous and extensive investigation of lower altitude weather using heterogeneous UAS. The immense scale, duration and millions of data points collected demonstrate the capacity of our algorithm to deploy heterogenous robots over hundreds of square kilometers, investigating and mapping meteorological data with a speed and resolution unmatched by UAS whose autonomy is limited to flying pre-planned profiles and transects. This represents a large advance in the state of the art.

This work is supported by NSF award #1539070 (Unmanned Aircraft System for Atmospheric Physics).



**Figure 8:** Plots for a representative autonomous flight are on the left; a preplanned profile flight is on the right. Measured temperature change over time ( $\frac{\delta F}{\delta t}$ ) is in the bottom row; the top row collects these data into a histogram for information gain computation.

## REFERENCES

- [1] Waseem Abbas and Magnus Egerstedt. 2012. Distribution of agents with multiple capabilities in heterogeneous multiagent networks - A graph theoretic view. *Proceedings of the IEEE Conference on Decision and Control (CDC)* (2012).
- [2] Waseem Abbas and Magnus Egerstedt. 2014. Characterizing Heterogeneity in Co-operative Networks From a Resource Distribution View-Point. *Communications in Information and Systems (CIS)* (2014).
- [3] Zoltán Beck, Luke Teacy, Alex Rogers, and Nicholas R Jennings. 2016. Online planning for collaborative search and rescue by heterogeneous robot teams. In *Autonomous Agents & Multiagent Systems (AAMAS)*.
- [4] Jesse Butterfield, Odest Chadwicke Jenkins, and Brian Gerkey. 2008. Multi-robot Markov random fields. In *Autonomous Agents and Multi-Agent Systems (AAMAS)*.
- [5] Ji Chen, Salar Moarref, and Hadas Kress-Gazit. 2018. Verifiable Control of Robotic Swarm from High-level Specifications. In *AAMAS*.
- [6] Jeff Craighead, Robin Murphy, Jenny Burke, and Brian Goldiez. 2007. A survey of commercial & open source unmanned vehicle simulators. In *Robotics and Automation, 2007 IEEE International Conference on*. IEEE.
- [7] Christopher Crick and Avi Pfeffer. 2003. Loopy belief propagation as a basis for communication in sensor networks. In *Uncertainty in Artificial Intelligence (UAI)*.
- [8] Yu Fu, Xiang Yu, and Youmin Zhang. 2015. Sense and collision avoidance of Unmanned Aerial Vehicles using Markov Decision Process and flatness approach. In *International Conference on Information and Automation (ICIA)*.
- [9] Seng Keat Gan, Robert Fitch, and Salah Sukkarieh. 2012. Real-time decentralized search with inter-agent collision avoidance. In *International Conference on Robotics and Automation (ICRA)*.
- [10] John Roy Garratt. 1994. The atmospheric boundary layer. (1994).
- [11] Nathan Koenig and Andrew Howard. 2004. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*. IEEE.
- [12] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. 2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory* 47, 2 (2001).
- [13] Pablo Lanillos, Seng Keat Gan, Eva Besada-Portas, Gonzalo Pajares, and Salah Sukkarieh. 2014. Multi-UAV target search using decentralized gradient-based negotiation with expected observation. *Information Sciences* 282 (2014).
- [14] Shayegan Omidshafiei, Christopher Amato, Miao Liu, Michael Everett, Jonathan P. How, and John Vian. 2017. Scalable accelerated decentralized multi-robot policy search in continuous observation spaces. In *International Conference on Robotics and Automation (ICRA)*.
- [15] Judea Pearl. 1988. *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann.
- [16] Michal Pechoucek and David Sislak. 2009. Agent-based approach to free-flight planning, control, and simulation. *Intelligent Systems, IEEE* 24, 1 (2009).
- [17] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. 2009. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*.
- [18] Alessandro Renzaglia, Christophe Reymann, and Simon Lacroix. 2016. Monitoring the evolution of clouds with UAVs. In *International Conference on Robotics and Automation (ICRA)*.
- [19] Christophe Reymann, Alessandro Renzaglia, Fayçal Lamraoui, Murat Bronz, and Simon Lacroix. 2017. Adaptive sampling of cumulus clouds with UAVs. *Autonomous Robots* (2017).
- [20] Radu Bogdan Rusu, Alexis Maldonado, Michael Beetz, and Brian Gerkey. 2007. Extending Player/Stage/Gazebo towards cognitive robots acting in ubiquitous sensor-equipped environments. In *ICRA Workshop for Networked Robot Systems, 2007, Rome, Italy*.
- [21] Peter Stone and Manuela Veloso. 2000. Multiagent systems: a survey from a machine learning perspective. *Autonomous Robots* 8, 3 (2000).
- [22] Zachary N. Sunberg, Mykel J. Kochenderfer, and Marco Pavone. 2016. Optimized and trusted collision avoidance for unmanned aerial vehicles using approximate dynamic programming. In *International Conference on Robotics and Automation (ICRA)*.
- [23] Philip Twu, Yasamin Mostofi, and Magnus Egerstedt. 2014. A measure of heterogeneity in multi-agent systems. *Proceedings of the American Control Conference* (2014).
- [24] Ryan K Williams and Gaurav S Sukhatme. 2011. Cooperative multi-agent inference over grid structure Markov random fields. In *Intelligent Robots and Systems (IROS)*.
- [25] Blake Wulfe, Mykel J. Kochenderfer, Sunil Chintakindi, Sou Cheng Choi, Rory Hartong-Redden, and Anuradha Kodali. 2018. Real-time prediction of intermediate-horizon automotive collision risk. In *AAMAS*.