

Toward Haptic-Based Management of Small Swarms in Cordon and Patrol

C. Kristopher Alder[†], Samuel J. McDonald[‡],
Mark B. Colton[‡], and Michael A. Goodrich[†]

Abstract—Distributed teams of agents can provide robust solutions to many problems of interest, and allowing a human to influence and manage those agents can extend the range of problems that can be solved while improving the team’s efficiency. Within this context, it is interesting to develop methods for interaction that are intuitive and that utilize haptic interaction so that the human manager need not be “heads down” in a graphical user interface. This paper presents a set of agent control algorithms that yield useful team performance and enable haptic-based management of team behaviors. A preliminary demonstration of the system is also presented.

I. INTRODUCTION

There are many current and future scenarios in which a human must manage a team of air, ground, and humanoid robots (generically referred to as *agents* in this paper). These scenarios include wilderness search-and-rescue [1], rescue operations in buildings damaged by fire or earthquake [2], [3], searching of a building by law enforcement agencies [4], pollution monitoring and clean-up [5], and military patrol and cordon operations in an urban environment [6]. In each scenario, the agent team serves as an extension of the human’s ability to gather information in complex and often dangerous environments, and the tasks are often time-sensitive. Enabling the human operator to manage the agent team in an intuitive, effective, and time-efficient manner is therefore critical to the success of operations involving agent teams.

The state of the art in controlling autonomous agents (in use by current military, law enforcement, and search-and-rescue agencies) is for a single agent to be controlled and monitored by one or many human operators (see, for example, [7]). This interaction model is clearly not ideal if the objective of employing autonomous agents is to augment the capabilities of humans and maximize the information-gathering capabilities of the team [8], [9]. A preferable interaction model is for a single human operator to control multiple autonomous agents. The effectiveness of such an approach is limited by the human’s ability to command the actions of multiple agents and receive information about the state of the agent team. When the agent team possesses appropriate autonomy for the given scenario, the problem becomes one of “team management” rather than “agent control,” enabling the human to focus on

task objectives and interpretation of gathered data, rather than on the agents.

In this paper we describe initial work on an approach to managing a team of agents by a single human user. Our focus is on the patrol and cordon scenario, although the approach will be generalizable to other scenarios where humans must control the movement of teams of autonomous agents, as well as the distribution of the agents within the team and relative to features in the environment. In a patrol and cordon scenario, an agent team works in an urban environment to search buildings. In *surround* mode the user selects a building or buildings for the team to surround, and the team autonomously forms the cordon. The user can see and feel the team distribution, and modify the distribution based on human knowledge of likely entrance and exit points, traffic patterns, obstructions, etc. The user has the ability to modify the size and shape of the cordon. When the user selects a new building or buildings to surround, the team disengages from the original building, autonomously switches into a *travel* mode, and re-forms around the new target.

The approach includes two elements: (1) swarming behaviors that enable autonomous agent teams to self-form around environmental features (buildings) and into travelling formations, and (2) graphical and haptic interactions that enable the user to see, feel, and command the team’s location and distribution. The swarming behavior is based on a distributed algorithm that governs the position of and communication between a team of 10–15 agents such that there is minimal need for globally shared information. Graph theory is used to describe relationships between the agents, with each agent corresponding to a node in the graph. During *surround* mode, the agent graph’s connections form a spanning ring, whereas the graph contains a spanning forest during *travel* mode, with each agent being influenced only by a single leading agent. While in *surround* mode the user can feel haptic feedback as he stretches, compresses, or reshapes the spanning ring. In *travel* mode haptic feedback is determined solely by the shape of the spanning forest during movement, so that the user may gain any intuitive sense for the location and overall shape of the team.

The objective of the present work is to create individual and team autonomy behaviors, as well as user interface methods, to enable a single user to manage a planar, swarm-like team in an urban patrol and cordon scenario. Although the work we present is novel, it is not without precedent. Decentralized unmanned ground vehicle (UGV) swarms have been controlled

[†]C. K. Alder and M. A. Goodrich are with the Computer Science Department at Brigham Young University

[‡]S. J. McDonald and M. B. Colton are with the Mechanical Engineering Department at Brigham Young University

successfully using haptic feedback from a bird’s eye view, to maximize manipulability or coverage [10], [11], [12]. In each of these studies, the primary objective was to improve operator performance by decreasing collisions without increasing operator workload. Results from user studies showed a decrease in task completion time and an increase in team coverage when using haptic feedback. Our work complements this prior work by identifying agent control algorithms that yield, by design, global behaviors that are amenable to haptic interactions.

II. INDIVIDUAL BEHAVIORS

In this section, we present the model by which each individual agent in the collective governs its motion. We restrict attention to agents that use as little centralized information as possible. Furthermore, we restrict attention to agents that can sense only two types of objects in the world: other agents and obstacles/points of interest. Based on sensor readings from these types of objects, each agent computes a force that pushes or pulls it in a direction that (a) keeps it close (but not too close) to its neighbors and (b) either keeps it in proximity to the boundary of an object if the agent is in a *surround* mode or helps it avoid an object if the agent is in *travel* mode.

A. Neighbour Selection

In the literature on bio-inspired collectives, there are two main methods by which an agent determines who influences it [13]: a metric-based method and a k-nearest-neighbors method. The metric-based method is typified by both Couzin’s model [14] and Reynolds’ model [15] in which agents are influenced by all neighbors within a radius of attraction, repulsion or orientation. The key idea in metric-based methods is that all agents within a certain distance influence the agent, regardless of how many agents are within those neighborhoods.

By contrast, the k-nearest-neighbors (NN) method assumes that each agent can only track a certain number of neighbors, and assumes that the neighbors that are nearest the agent are the ones most likely to exert influence. The NN method is typified by Ballerini’s work in which evidence suggested that this model was both biologically plausible and better able to explain the behaviors of starlings than a metric-based method [16].

We adopt the NN method because prior work suggests that it produces collectives that are less likely to fragment in the presence of large perturbations [17]. This means that the forces that an agent experiences are determined by a limited number of agents. This induces what we call an *interagent influence topology*. Later in this paper, we will consider a different topology, which we call the *communication topology* that may allow an agent to send and receive messages from other agents even if those other agents do not directly influence the forces used by the agent to select an action.

Let k denote the number of other agents that an agent is influenced by. Both empirical evaluations and previous work [18] indicate the following trends: larger values of k will cause the swarm to settle on a solution more quickly,

while lower values produce swarms that are more flexible and responsive to control by human operators.

It is straightforward to create an algorithm in which each agent identifies its k -nearest neighbors. Let \mathbf{x}_i denote the position of the i^{th} agent in the collective and let $N(\mathbf{x}_i; k)$ denote the positions of the k -nearest neighbors of agent i . From \mathbf{x}_i and $N(\mathbf{x}_i; k)$ distances between an agent and each of its neighbors are easily computed. Let \mathbf{X}_i denote the matrix constructed from $N(\mathbf{x}_i; k)$ by sorting the neighbor set from nearest to farthest from the agent. Thus,

$$\mathbf{X}_i = [\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_k] \quad (1)$$

where we assume that each \mathbf{x}_i is a column vector and \mathbf{x}_1 is the neighbor nearest to agent i , \mathbf{x}_2 is the next closest, and so on.

B. Obstacle Sensing

Recall that we assumed each agent could sense both other agents as well as the positions of obstacles or points of interest in the world. We assume a radar or lidar-type sensor that is able to measure both distance to and direction of these objects. This sensor is consistent with our goal to minimize centralized information in the world because the agents need not understand the location and shape of, for example, a building in the world as long as they can distinguish between a distance measure from a neighboring agent and a distance from an obstacle. As we shall demonstrate, this design frees the user from the need to communicate the location and shape of buildings to each of the agents. It also allows the agents to respond to changing conditions without directly involving the human operator.

In our simulations, each agent sends out radar “pings” at regular intervals in a 360-degree arc. Larger objects, such as walls, will be detected by multiple of these simulated radar “pings”. This yields a set of readings of obstacle locations uniformly distributed around agent i , $\{\xi_i^j : j \in \{1, 2, \dots, \frac{360}{\Delta} - 1\}\}$ where Δ is the angle between the different pings. Note that we are using a convention where the symbol for agent-related parameters uses a letter from the English alphabet while the symbol for the corresponding object-related parameter uses the corresponding letter from the Greek alphabet.

We sort these readings from nearest to farthest and to drop readings that are beyond the range of the sensor. This sorted vector of readings is denoted by the array Ξ_i . Because objects outside of the sensors’ range may not be detected at all, the size of the array may vary as agents move in the world and objects enter and leave sensor range. Thus,

$$\Xi_i = [\xi_i^1 \xi_i^2 \dots \xi_i^n] \quad (2)$$

where $n < \frac{360}{\Delta}$ is the number of sensor readings within sensor range and ξ_i^1 is the location of the sensor ping closest to agent i , ξ_i^2 the next closest reading, and so on..

C. Emphasizing Unique Influences

The first important factor in determining how objects and other agents influence an agent, and an innovative part of the

agent controller, is that neighbors or sensor readings that are “unique” should have stronger influence. More specifically, our model limits the amount of influence a small cluster of agents or sensor pings can exert, which tends to keep the interagent topology connected, making it robust as the collective moves through the world or encounters a building.

For each agent $a \in N(\mathbf{x}_i, k)$, we assign a weighting w_i^a which is a function of how unique agent a is relative to other agents. For the nearest agent, which is identified by $a = 1$ since we sorted agents from nearest to farthest, let $w_i^1 = 1$. This initializes the weight-assignment algorithm. For each other agent $a \in \{2, \dots, k\}$, let w_i^a be given by

$$w_i^a = \frac{1}{\pi} \min_{1 \leq b < a} \cos^{-1} \left(\frac{\mathbf{x}_a^T \mathbf{x}_b}{\|\mathbf{x}_a\| \|\mathbf{x}_b\|} \right) \quad (3)$$

This equation iteratively assigns a weight to every agent $a \in \{2, \dots, k\}$ proportional to the smallest angle between agent a and every other agent that has previously been assigned a weight. The idea is that the normalized inner product given by the fraction gives the cosine of the angle between unassigned agent a and every previously assigned agent b . The arc cosine gives the angle between agent a and agent b , and the smallest such angle indicates how directionally similar that agent is to every previously assigned agent. If there are other, closer agents in the general direction of agent a then agent a is in a cluster of agents and shouldn't receive a high weighting. The $\frac{1}{\pi}$ normalizes each weight so $w_i^a \in [0, 1]$. The weights are then normalized so that they sum to 1.

Stated simply, agents that lie in a very similar direction to other neighbouring agents are given lower influence.

The same algorithm can be applied to each column in ξ_i yielding a set of “uniqueness” weights for each ping, ω_i^j where $j \in \{1, 2, \dots, n\}$.

D. Calculating Force Strength

The weights given by the algorithm described in the previous section determine *how much* an agent in agent i 's neighborhood and how much obstacles will influence agent i . We now need to determine *how* agent i 's behavior should be influenced by those agents. The introduction identified the second important factor in determining the strength of interaction, namely that, depending on mode, agents want to be close but not too close to other agents and to objects in the world.

Let d_i and δ_i denote the ideal standoff distance that agent i wants to maintain between agents and obstacles, respectively. Given these ideal standoff distances, we can determine how influences from other agents and obstacles push or pull an agent to a new location.

If a neighbor of agent i is too far away then agent i should be attracted to that neighbor. Formally, agent i is attracted to neighbor j when $\|\mathbf{x}_i - \mathbf{x}_j\| > d_i$. We let the strength of the attractive influence grow as the distance increases. Similarly, agents that are closer than the stand-off distance d_i exert a repulsive force that approaches infinity as the distance between the two agents approaches zero. This prevents two agents from

colliding. The strength of the force from agent j to agent i is given by

$$s_i^j = w_i^j \left| \frac{1}{d_i} - \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|} \right| \quad (4)$$

Future work will consider refinements of this equation, but the principles behind it are simple and are consistent with both Couzin's model [14] and potential-field methods for agent control [19]. The idea is that the repulsive force increases to infinity as the objects approach (as described in the text above) and attractive forces increase as distance increases.

The radar “pings” represented by Ξ_i exert similar repulsive and attractive forces on agent i with strengths scaled by ω_i . This yields the strength of the force from sensor reading j on agent i as

$$\sigma_i^j = \omega_i^j \left| \frac{1}{\delta_i} - \frac{1}{\|\mathbf{x}_i - \xi_j\|} \right| \quad (5)$$

E. Agent Control

Agents have two different modes or states: *surround* and *travel*. As described in the next section, agents use a simple form of distributed decision-making to determine when to switch behaviors, and future work should explore other methods (see [20], [21]).

The direction that the agent should travel is given by the sum of the forces derived from other agents and from objects in the world. The previous sections have provided definitions that apply to the most general situation, but some forces are ignored in some of the modes. The most general situation is when the agents are in *surround* mode, in which case the forces on the agent are given by:

$$\begin{aligned} \Delta \mathbf{x}_i &= \sum_{j=1}^k s_i^j \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|} \left(\text{sgn}(d_i - \|\mathbf{x}_j - \mathbf{x}_i\|) \right) \\ &+ \sum_{j=1}^n \sigma_i^j \frac{\xi_j - \mathbf{x}_i}{\|\xi_j - \mathbf{x}_i\|} \left(\text{sgn}(\delta_i - \|\xi_j - \mathbf{x}_i\|) \right). \end{aligned}$$

This equation takes a summation over all neighbors (the first line) and all objects (the second line) of a set of unit vectors weighted by the strengths of the objects/neighbors. The unit vectors point in the direction of neighbors/objects, and the sign of these vectors is determined by whether the objects/neighbors attract or repel the agent.

In *travel* mode, two changes are made to this equation: First, only one other agent is considered, a special agent that is leading the group. Second, the agent is never attracted to an object, only repelled. This yields

$$\begin{aligned} \Delta \mathbf{x}_i &= s_i^\ell \frac{\mathbf{x}_\ell - \mathbf{x}_i}{\|\mathbf{x}_\ell - \mathbf{x}_i\|} \left(\text{sgn}(d_i - \|\mathbf{x}_\ell - \mathbf{x}_i\|) \right) \\ &+ \sum_{\{j: \|\xi_j - \mathbf{x}_i\| < \delta_i\}} \sigma_i^j \frac{\xi_j - \mathbf{x}_i}{\|\xi_j - \mathbf{x}_i\|} \left(\text{sgn}(\delta_i - \|\xi_j - \mathbf{x}_i\|) \right) \end{aligned}$$

where ℓ indicates the index of the leader agent.

Observe that $\Delta \mathbf{x}_i$ is controller agnostic, meaning that we can use this signal as the input to a PD or other controller that causes the agent to track toward the desired direction.

This provides flexibility in applying the method to many different kinds of agents. In the simulations presented below, we assume a robot capable of omnidirectional travel, such as a quadcopter, and assume that the changes in direction are the commanded directions to the agent. The results shown in the demonstration below should generalize to other kinds of agents and controllers provided that the controller is able to quickly track the desired direction.

III. COLLECTIVE BEHAVIORS

The purpose of designing the interactions between individual agents and their environment is to produce useful behaviors for the swarm as a whole. We adopt a graph theoretic approach to describing and analyzing how the individual agent interactions produce a global behavior, following the example in [22]. Stated simply, viewing the swarm’s interactions as a graph allows us to reason about these emergent behaviors. Each agent can be considered a node in the graph, while neighbouring agents’ influence can be represented as directed connections between those nodes.

As noted in the introduction, there are two different graphs used in the work: an inter-agent influence graph, which is the graph used to compute the forces experienced by the agent in the previous section, and a communication graph, which we now describe. Both graphs use the same set of vertices, but their edge sets may differ. In this paper, we assume that the two graphs are the same for simplicity, but future work should explore different topologies because detecting the set of nearest neighbors and the set of neighbors with whom an agent can communicate can be radically different. Consider, for example, a neighborhood topology obtained using visual sensing and a communication topology obtained using power-limited broadcast mechanisms; these two topologies would be very different.

A. Selecting Desired Inter-agent Distances

It is desirable to create collective behaviors that are robust to different physical scales and different numbers of agents. This presents a challenge for a group of agents that want to surround a building, because the spacing between agents depends on the size of the building and the number of agents.

In the spirit of distributed computing, we use a consensus algorithm for agents to reach an agreed up on ideal standoff distance d^* , yielding, for all agents i , $d_i \rightarrow d^*$. Each agent calculates the average distance of its neighbours and communicates this to each of the neighbours. It then increases its own value for d_i if the averages of its neighbours are larger, and decreases d_i if it is lower. This is precisely the Laplacian consensus algorithm described in [22], [23], where state is the average distance between an agent and its neighbors. This has the effect of allowing the agents to determine the ideal inter-agent distance without user input. It is well established that as long as the graph of agents is connected, all agents will eventually converge on the same value [22], [23].

B. Switching Between Modes

Using the communication topology, agents communicate commands they receive from a human operator to switch between *surround* and *travel* modes. This allows the human user to control the entire team while only communicating directly with a small number of agents. The algorithm by which this occurs has two parts, one part that applies to agents who receive direct commands from a human operator, and a second part for the agents with whom those commanded agents interact. Consider first the agents that have been commanded to travel in a new direction. These agents switch to the *travel* mode and begin travelling in the given direction. These agents remain in *travel* mode until they have travelled the distance specified by the human operator. Upon reaching their destination or when the human cancels the command, they then change back into *surround* mode.

Consider now the agents that do not receive a direct command from the human. These agents watch their k selected neighbours. When an agent detects that one of its neighbours has changed into *travel* mode, it also changes into *travel* mode and sets the neighbour in *travel* mode to become its leader (if more than one neighbor is in *travel* mode, one is selected randomly). Agents remain in *travel* mode until they detect that their leader has changed back to *surround* mode.

In practice the two parts of this algorithm mean that the human induces a change to the entire group by selecting one or a few leaders and then letting the mode changes in those leaders propagate through the group. As long as the group stays connected, all agents will eventually switch to the *travel* mode and back to the *surround* mode.

C. Spanning Ring in Surround Mode

We are now in a position to precisely define the swarm-level behavior that we want to occur when the agents are in the different modes. We begin with *surround* mode.

The desired group configuration in *surround* mode is for the agents to distribute themselves around the perimeter of a building at equal inter-agent distances. This can be associated with a global property of the inter-agent graph. Formally, when the agent teams come to a stable position around a building, there should exist a subset of the agent graph’s connections that form a *spanning ring* around the building. Using terminology from graph theory [24], we define a spanning ring as sub-graph that has a planar embedding (has no intersecting edges) which spans the entire graph (includes all agents) and such that each vertex has exactly two neighboring vertices. The existence of a spanning ring sub-graph indicates that the agent team has successfully surrounded the building.

Importantly, the spanning ring is a global property that communicates the overall shape of the agents’ formation to the human operator in an intuitive way. We discuss this further below.

D. Spanning Forest in Travel Mode

While the agent team is travelling between locations, each agent in the graph will be in *travel* mode. Except for the agents

directly influenced by the human, each agent has identified exactly one leader agent to follow. Thus, the graph consists of multiple trees, one tree per agent under control of the human. Each tree is a *spanning tree* of the associated sub-graph of all agents connected to the same leader.

Thus, the global characteristic associated with the *travel* mode is a *spanning forest* sub-graph that consists of one or more spanning trees of agents. Metaphorically, this can be thought of as a “flock of flocks” or as the “vee of vee” that can be seen in nature and in WWII aerial tactics. Since all agents in the swarm are either leaders or are covered by exactly one spanning tree, the existence of the spanning forest indicates that the agent team has successfully adopted the *travel* mode.

As with the spanning ring, the spanning forest is a global property that communicates the overall shape of the agents’ formation to the human operator in an intuitive way. We discuss this below.

IV. HAPTIC INTERFACE

The challenge for the human is to manage the team of autonomous agents effectively, without being overburdened or losing situational awareness through “heads-down” attention focused on a graphical user interface. In the present work, we use a visual display augmented with a haptic interface to represent the team and its environment to the user and to enable the user to command new locations and distributions to the swarm. The haptic element is accomplished by creating virtual deformable volumes that enclose the spanning ring and spanning forest during the *surround* and *travel* modes, respectively. The user can feel, reposition, and deform these volumes, with corresponding changes in the location and shape of the team and in the distribution of the agents within the team. We present details of our approach in this section.

A. Modeling Clay: A Haptic Metaphor

Based on the results of a brainstorming breakout session at the 2012 AAAI Fall Symposium on Human Control of Biological Swarms, Diana et al. proposed the idea of using a deformable medium, such as modeling clay, as a “joystick” to command the distribution of large-scale swarm-like teams of homogeneous vehicles [25]. They demonstrated a molding scheme in which an operator formed modeling clay into various shapes in the view of an overhead camera and a team of micro robots replicated the formation commanded by the shaped clay. We modify the modeling clay metaphor so that a human can shape the distribution of agent teams by manipulating a *virtual* deformable volume through stretching, pulling and other operations. The modeling clay metaphor forms the basis for the haptic (force feedback) sensations that the user feels while distributing the agent team. Note that, unlike the work in [25], physical modeling clay is not used in our method; the concept and physics of modeling clay are used to generate the visual and haptic representation of the agent team.

We create a discrete approximation of the continuous clay metaphor by introducing potential force field spheres at the

location of each of the agents in the spanning ring and at multiple points between agents. In essence, these potential spheres form the nodes of a “force graph” on which the haptic interaction forces and graphical representation of the deformable volume (“virtual modeling clay”) are based.

Movement of the haptic interface in 3D space maps to movement of a graphical proxy on the display screen. The user exerts an external force, \mathbf{f}_s , on the network by moving the haptic proxy toward and into the outer boundary of one or more of its component spheres, as shown in Figure 1. The force \mathbf{f}_s is calculated as a linear function of the penetration vector \mathbf{e} , $\mathbf{f}_s = k_s \mathbf{e}$, where k_s is the stiffness constant of the sphere. Deformation of the network occurs under the action of the applied force \mathbf{f}_s . The haptic feedback force felt by the user in response to the interaction is then $\mathbf{f}_d = -\mathbf{f}_s$.

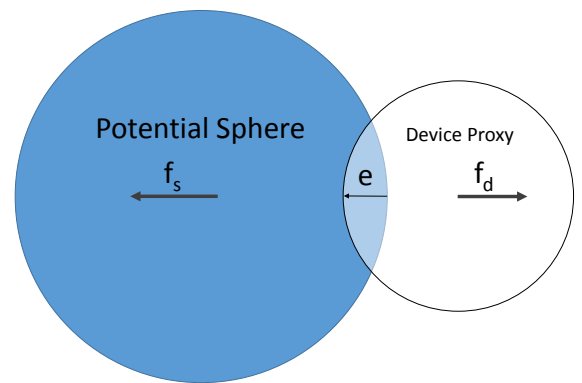


Fig. 1: Haptic device proxy (white) in contact with a spherical potential force field (blue). \mathbf{f}_s is the force the user exerts on the network of spheres and \mathbf{f}_d is the return force felt by the user.

For force calculation purposes, the potential spheres also form the mass nodes of a virtual mass-spring-damper network. This dynamic network allows the model to update according to current team distribution and accept input forces from the operator to manipulate the team. To prevent the device proxy from passing between neighboring nodes, they are positioned a distance

$$\delta_{spacing} = r_n / n_{spacing}$$

apart, where r_n is the node radius and $4 \leq n_{spacing} \leq 6$. Choosing appropriate values for mass, spring, and damping constants allows each node to maintain sufficient distance relative to neighboring nodes, stabilize the model and create a distinguishable volume with which to interact.

The collective shape of the deformable volume should be a simple topology that the operator understands. To communicate a coherent group of agents in either *surround* or *travel* mode, a deformable ring is chosen as this topology, and is formed by connecting nodes that span across agent locations. The agent locations that define the deformable ring shape are determined by the current mode, and will be discussed in the following section. The deformable ring forms the basis for computing the haptic feedback force felt by the operator.

B. Mapping of Agent Behaviors to Haptic Feedback

After the agent team has successfully surrounded a desired building or location in the environment, the operator may wish to explore the current state of the team. As mentioned above, a virtual deformable ring is created from the spanning ring formed by the agents. The resulting force graph provides the interaction forces for commanding and receiving feedback from the distributed agents. Placing additional nodes between agent locations ensures that the user can interact with a seemingly continuous deformable volume, rather than with discrete agents only.

While exploring the current distribution of the team, the operator receives force feedback when in contact with the volume. For a device proxy in contact with N nodes, the total haptic feedback force felt by the operator is

$$\mathbf{f}_h = \sum^N \mathbf{f}_d + \mathbf{f}_g,$$

where \mathbf{f}_d represents the haptic force from a single node, as described previously, and \mathbf{f}_g is a ground reaction force felt when the user comes in contact with the virtual ground. In addition to exploration, the user may also manipulate the distribution of the team. This is done by prodding the ring from the inside or outside, causing the volume to deform and propagate movement to each of the individual agents, as discussed in previous sections.

While the team is in motion from one location to another, the user may desire to feel the overall shape of the team while traveling as a spanning forest. For this purpose, a convex hull [26] of agent locations is computed based on their global positions in the environment. A *travel*-mode virtual deformable ring is formed by using the agents which are located on the edge of the convex hull as an ordered list of positions. Because the user would not benefit from feeling the inside of the convex hull, we create a distinct enclosed deformable volume. When the device proxy comes within the 2-dimensional plane of nodal movement and is inside the convex hull, the user feels an out-of-plane (vertical) feedback force, \mathbf{f}_c , which creates a virtual surface to differentiate the team formation from the ground plane. The total haptic feedback force felt by the operator while in *travel* mode is

$$\mathbf{f}_h = \sum^N \mathbf{f}_d + \mathbf{f}_g + \mathbf{f}_c,$$

and represents the forces felt as the operator comes in contact with the set of nodes comprising the ring, the ground reaction force, and the convex hull force.

V. SIMULATION SYSTEM AND DEMONSTRATION

The graphical interface and haptic feedback of the agent team and its environment were programmed using CHAI 3D, “an open-source set of C++ libraries for computer haptics, visualization and interactive real-time simulation” (www.chai3d.org/concept.html). Figure 2 shows an operator’s view of a simulation system that we have developed. The

surrounded building has been chosen as a desired target by the user. From this *surround* state, the operator may manipulate the current distribution by interacting with the deformable ring (hidden from the user in this figure), or command the team to travel to one of the other buildings. A travel direction, \mathbf{d}_t , may be chosen by holding a button on the haptic device end effector. The operator holds the button down when close to a set of agents, drags the cursor in a specified direction and releases the button to commence movement. This feature allows the human operator to use global information to specify travel directions with respect to the current location of the team. While the button is held down, an addition travel force, $\mathbf{f}_t = -k_t \mathbf{d}_t$, is applied to the haptic device, where k_t is a stiffness parameter. This travel force serves to alert the user where the current location of the team is relative to the commanded location. The simulation system will be used to test proposed agent behaviors, feedback algorithms and user interface capability as an integrated design.

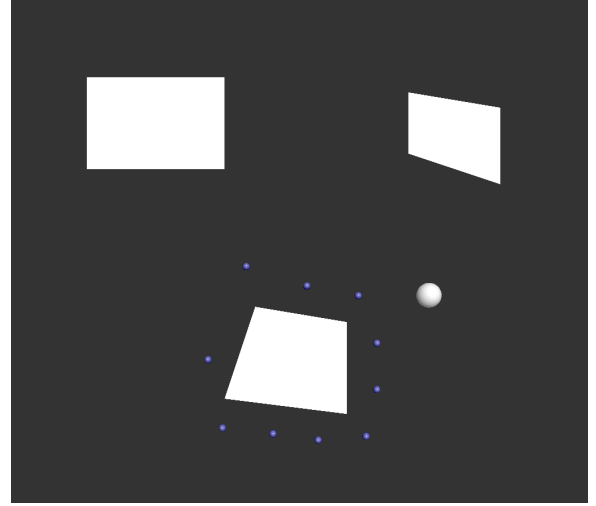


Fig. 2: The human operator has global visibility of the team and environment from a bird’s-eye view. A collection of buildings (white) are shown in the environment as candidate targets of interest. The team of agents (blue dots) are shown in *surround* mode around a building. The haptic proxy is the white sphere.

Figure 3 illustrates how user input via the haptic interface maps to commands to the autonomous team of agents. In this simulation there are 10 agents, although the methods presented in this paper are theoretically scalable to large numbers of agents. Empirically, values of $2 \leq k \leq 5$ tended to produce cohesive swarms and avoided fragmenting, where k is the number of other agents that an agent is influenced by, as described above. Results shown here are for $k = 5$. Sub-figures (a) and (b) show the team in *surround* mode around a building. The haptic proxy (white sphere) is not interacting with the team at this point in time. Sub-figure (a) shows the autonomous agents, whereas sub-figure (b) shows the deformable volume formed from the spanning ring computed from the agent distribution. In *surround* mode the operator may explore the state of the team and receive force feedback based on the current distribution. Sub-figures (c) and (d) show

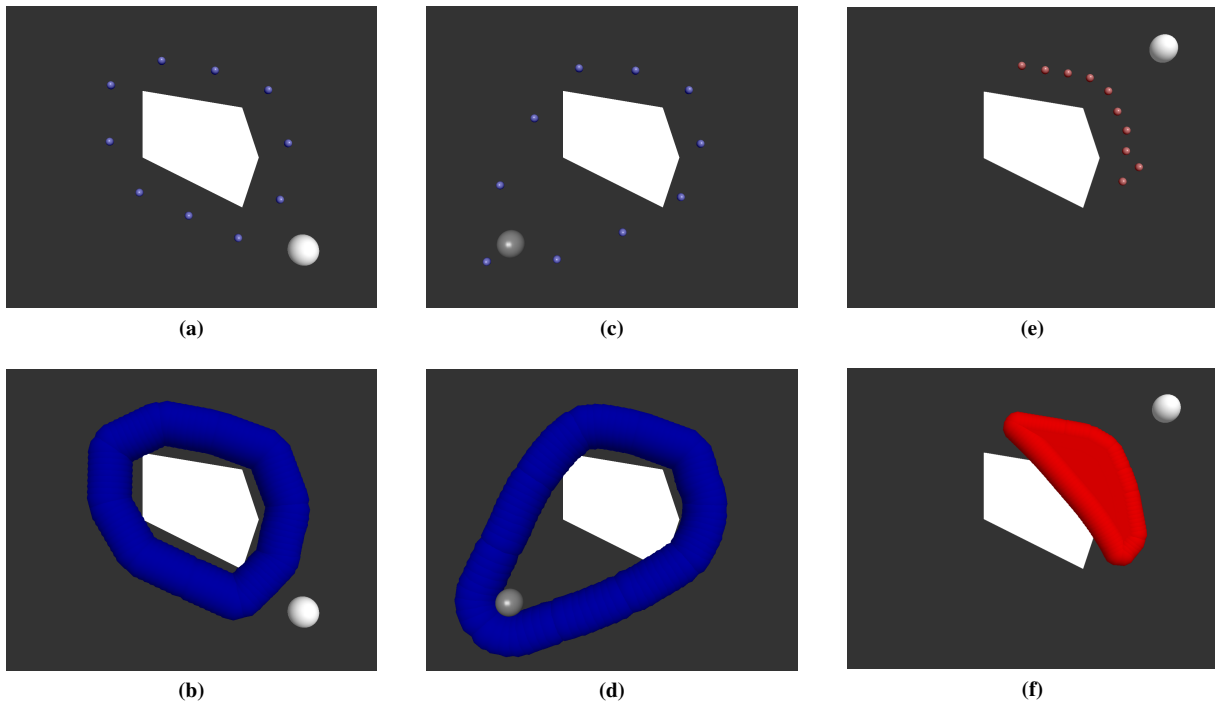


Fig. 3: Demonstration of agent behaviors and haptic interactions. (a) The agent team autonomously surrounds a building geometry (shown in white). (b) A deformable surrogate is formed to intuitively sense the ring topology and team collective behavior. (c) External forces in the surrogate model propagate to movements by individual agents. (d) The human operator may manipulate the team distribution by pulling the volume in a desired direction. (e) The operator commands the team to travel in a desired direction. (f) The outer, convex hull shape is formed to provide haptic feedback based on tree topology.

the operator changing the team distribution via the haptic proxy through stretching or pulling actions. When the operator manipulates the deformable volume, it influences the collective team behavior and individual agents adjust in a distributed way. Sub-figures (e) and (f) show the operator commanding the team to travel in a desired direction. During this movement, the operator may explore the overall shape of the spanning forest by interacting with the top virtual surface of the formation.

VI. SUMMARY AND FUTURE WORK

In this work we have begun to develop methods to enable autonomous patrol and cordon by a distributed team of agents in an urban environment under the supervisory management of a single human user. We have developed *surround* and *travel* swarm behaviors that enable the team to autonomously form a cordon around a building and travel to a new location, respectively, where the building and new location can be selected by the user. Input and force feedback methods have been presented that allow the user to change and feel the distribution of the swarm via a haptic interface.

There are a number of open questions that have not been addressed in this demonstration. Many of these have been identified in the body of the paper, but three deserve to be emphasized in this section. First, although the demonstration showed that the algorithms controlling agent behavior tend to produce desirable team-level behavior, we have not presented an analysis of the conditions that guarantee these team-level behaviors. This problem is very important because without understanding when the global behaviors are produced, it is

impossible to have bounds on how well the algorithms afford human interaction.

Second, evidence needs to be gathered that the system works with real humans using real robots in real-world settings. A series of user studies and demonstrations with physical robots is needed to gain confidence that the system could be useful for the problems mentioned in the introduction.

Third, the demonstration was only for a fixed number of agents. One of the desirable theoretical properties of a set of distributed agents is that their behavior is robust when some agents make errors and others are lost. The individual agent algorithms are designed to scale to more agents (e.g., using consensus to identify the desired inter-agent spacing) and the global properties exploited by the haptic controller should be invariant to the number of agents (e.g., the spanning ring and spanning forest), but evidence is needed to support this claim.

VII. ACKNOWLEDGMENTS

This work was funded by the United States Army Research Laboratory through contract W911NF-14-1-0633. All statements, findings, and results are the responsibility of the authors and may not reflect the opinions of the funding organization.

REFERENCES

- [1] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey, "Supporting wilderness search and rescue using a camera-equipped mini UAV," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 89–110, 2008.
- [2] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *Pervasive Computing, IEEE*, vol. 3, no. 4, pp. 24–33, Oct 2004.

- [3] L. Alboul, J. Saez-Pons, and J. Penders, "Mixed human-robot team navigation in the GUARDIANS project," in *Proceedings of the International Conference on Safety, Security, and Rescue Robotics*, Sendai, Japan, October 2008.
- [4] H. Jones and P. Hinds, "Extreme work teams: Using swat teams as a model for coordinating distributed robots," in *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW '02. New York, NY, USA: ACM, 2002, pp. 372–381. [Online]. Available: <http://doi.acm.org/10.1145/587078.587130>
- [5] S. Jung and M. A. Goodrich, "Shaping couzin-like torus swarms through coordinated mediation," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1834–1839.
- [6] P. Maxwell, A. A. Maciejewski, H. J. Siegel, J. Potter, and J. E. Smith, "A mathematical model of robust military village searches for decision making purposes," in *IKE*, 2009, pp. 311–316.
- [7] R. R. Murphy and J. L. Burke, "The safe human-robot ratio," in *Human-Robot Interaction in Future Military Operations*, M. Barnes and F. Jentsch, Eds. Ashgate Publishing, 2010, ch. 3, pp. 31–49.
- [8] M. A. Goodrich, "On maximizing fan-out: Towards controlling multiple unmanned vehicles," in *Human-Robot Interactions in Future Military Operations*, M. Barnes and F. Jentsch, Eds. Surrey, England: Ashgate Publishing, 2010.
- [9] J. M. Whetten, M. A. Goodrich, and Y. Guo, "Beyond robot fan-out: Towards multi-operator supervisory control," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, Istanbul, 2010.
- [10] T. Setter, A. Fouraker, H. Kawashima, and M. Egerstedt, "Haptic Interactions With Multi-Robot Swarms Using Manipulability," *Journal of Human-Robot Interaction*, vol. 4, no. 1, p. 60, 2015. [Online]. Available: <http://humanrobotinteraction.org/journal/index.php/HRI/article/view/186>
- [11] S. Nunnally, P. Walker, M. Lewis, N. Chakraborty, and K. Sycara, "Using Haptic Feedback in Human Robotic Swarms Interaction," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57, no. 1, pp. 1047–1051, Sep. 2013.
- [12] S. Nunnally, P. Walker, N. Chakraborty, M. Lewis, and K. Sycara, "Using Coverage for Measuring the Effect of Haptic Feedback in Human Robotic Swarm Interaction," *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 516–521, Oct. 2013.
- [13] e. a. A. Strandburg-Peshkin, C. R. Twomey, "Visual sensory networks and effective information transfer in animal groups," *Current Biology*, vol. 23, no. 17, pp. R709–R711, 2013.
- [14] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and H. R. Franks, "Collective memory and spatial sorting in animal groups," *Journal of Theoretical Biology*, vol. 218, no. 1, September 2002.
- [15] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *ACM Siggraph Computer Graphics*, vol. 21, no. 4. ACM, 1987, pp. 25–34.
- [16] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini *et al.*, "Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study," *Proceedings of the national academy of sciences*, vol. 105, no. 4, pp. 1232–1237, 2008.
- [17] M. A. Goodrich, P. Sujit, S. Kerman, B. Pendleton, and J. Pinto, "Enabling human interaction with bio-inspired robot teams: Topologies, leaders, predators, and stakeholders," Brigham Young University, Tech. Rep. BYU-HCMI Technical Report 2011-1, 2011.
- [18] D. S. Brown, S.-Y. Jung, M. Goodrich *et al.*, "Balancing human and inter-agent influences for shared control of bio-inspired collectives," in *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4123–4128.
- [19] R. C. Arkin, *Behavior-Based Robotics*. Cambridge, Massachusetts: MIT Press, 1998.
- [20] D. J. T. Sumpter, *Collective Animal Behavior*. Princeton, N.J.: Princeton University Press, 2012.
- [21] T. D. Seeley, *Honeybee democracy*. Princeton University Press, 2010.
- [22] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.
- [23] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 1859–1864.
- [24] C. Godsil and G. F. Royle, *Algebraic graph theory*. Springer Science & Business Media, 2013, vol. 207.
- [25] M. Diana, J. P. De La Croix, and M. Egerstedt, "Deformable-medium affordances for interacting with multi-robot systems," *IEEE International Conference on Intelligent Robots and Systems*, pp. 5252–5257, 2013.
- [26] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*. Springer, 2000.