Two Algorithms for the Movements of Robotic Bodyguard Teams

Taranjeet Singh Bhatia, Gürkan Solmaz, Damla Turgut and Ladislau Bölöni
Department of Electrical Engineering and Computer Science
University of Central Florida
4000 Central Florida Blvd, Orlando FL 32816
{tsbhatia, gsolmaz, turgut, lboloni}@eecs.ucf.edu

Abstract

In this paper we consider a scenario where one or more robotic bodyguards are protecting an important individual (VIP) moving in a public space against harassment or harm from unarmed civilians. In this scenario, the main objective of the robots is to position themselves such that at any given moment they provide maximum physical cover for the VIP. The robots need to follow the VIP in its movement and take into account the movements of the civilians as well. The environment can also contain obstacles which present challenges to movement but also provide natural cover. We designed two algorithms for the movement of the bodyguard robots: *Threat Vector Resolution* (TVR) for a single robot and *Quadrant Load Balancing* (QLB) for teams of bodyguard robots. We evaluated the proposed approaches against rigid formations in a simulation study.

Introduction

With the advancements of hardware in robotic technology, security and military applications in robotics have been continuously growing due to the demand fueled by private and public investments. Recently, demonstrated capability of mobile robots by research teams such as Mars rover (Maurette 2003), BigDog (Raibert et al. 2008) have shown the potential for reducing the need of human presence for dangerous scenarios such as nuclear plant toxic waste cleanup, search and rescue missions, border patrolling and outer space exploration. One application of mobile robots is providing personal security and protecting the life of highly important individuals, formally known as close protection service. The nature of this challenging security task requires the human agents to identify and assess all threats or risks in the environment continuously. Additionally, this field comprises of various security sub-tasks such as surveillance, patrolling, path planning, formation maintenance and cooperative architecture.

This study is motivated toward the development of a close protection behavior in simulation. In this paper, we provided mathematical formulation of collaborative security by multiple robot bodyguards with implicit communication. Furthermore, we applied effective multi-robot positioning approach in order to minimize the potential exposure of the very important individual (VIP) to open space that any possible attacker may want to breach to hurt the VIP. By considering single robot, we propose an approach, which we named *Threat Vector Resolution* (TVR), based on the threat assessments using the threat vectors. For collaborative security with multiple robots, we proposed the *Quadrant Load Balancing* (QLB) approach, in which we aim to divide the load using quadrants and minimize the formation time of multiple robots considering the previous positions of the robots.

Related Work

Developing artificial close protection involves learning for various sub-tasks: leader-follow task, crowd interaction, localization and formation. The experiment related to follow a human agent while moving through dense crowd has been approached using machine learning methods such as SVM and neuro-evolution (Khan, Arif, and Bölöni 2014). Many studies related to security such as ARMOR (Pita et al. 2008), IRIS (Tsai et al. 2009), GUARDS (Pita et al. 2011), PROTECT (Shieh et al. 2012), TRUSTS (Yin et al. 2012), and RaPtoR (Varakantham, Lau, and Yuan 2013) consider placement of checkpoints and deployment of patrol teams to provide protection against probable attacks by terrorists and criminals. These studies have been approached the security problem using Stackelberg's Security Game theoretic approach. Most of the applications, which use Stackelberg Security Game, require generating mixed strategies for a group of defenders and adversaries over an exponential number of routes or schedules. This brings expensive computation costs for the autonomous robots which have limited computation, communication and power resources.

Another security-related problem which have been studied widely is multi-robot patrolling ((Agmon, Kaminka, and Kraus 2011) (Vanek et al. 2010) (Portugal and Rocha 2013) (Fazli, Davoodi, and Mackworth 2013)). Multi-agent based patrolling requires exponential decision making in order to minimize time lag between two visits of the agents to the same location or gain advantage over adversary by protecting a particular geographical area. This problem does not consider close encounter, escape or involvement strategies against adversary. Furthermore, this problem does not take the importance level of the target and impact of joint activi-

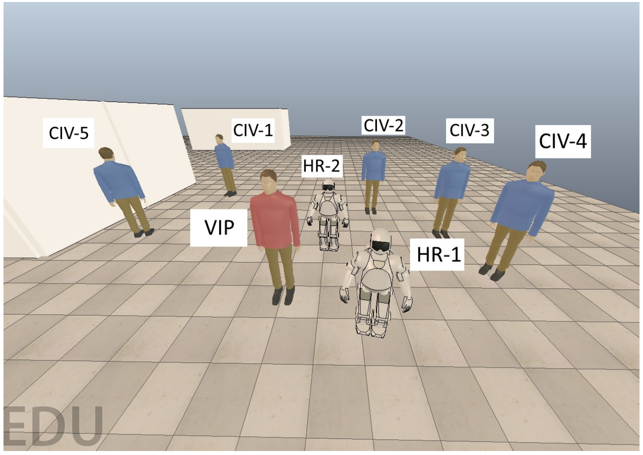

Figure 1: 3D visualization of the scenario with VIP, two humanoid robots displayed as HR-1, HR-2 and civilians displayed as CIV-1,2,3,4,5.

ties into consideration. The use of wireless sensor networks with mobile sinks are proposed for the event coverage problem, which is defined as providing security to people in large areas by covering possible incidents ((Solmaz and Turgut 2014) (Solmaz and Turgut 2013)). In this model, multiple mobile sinks patrol in their allocated regions and position themselves according to their distances to possible events and the movement of pedestrians. (Khan et al. 2012) provide modeling framework involving human-robot interaction for patrolling task in a cross-cultural environment. This framework captures and analyzes the behavioral perception of the actions of the soldier and the robot by the local population.

Current literature in the field of swarm robotics mostly focus on the cooperation and the collectivity of large number of robots as a group. The cooperation strategies (Cao, Fukunaga, and Kahng 1997) include formation control, foraging, docking, and flocking behavior. The approaches proposed in these studies focus on simple homogeneous behaviors of robots in a robot group rather than implementing an individual robot with a control mechanism to solve a complex task on its own.

Background on CPO Domain

The close protection operatives (CPOs) are team of individuals who are trained in close protection skills. Their skills involve protecting a person from both identified or unidentified threats and vital risks. In real world, it is impossible to keep risk free environment for a principal which is VIP at all times. Therefore, CPOs need to identify and assess all threats or risks in the environment continuously. At the same time, CPOs have to take any preventive measures to deviate or avoid any life threatening situation. The field of close protection is composed of different security aspects such as protective team formation, personal escort section, security advance party, residence security team, venue protection, protection principles and techniques, and counter action teams (Schneider 2009).

This paper deals with the modeling of robotic agents, which can be seen as bodyguards or individual team member of the personal escort section (PES) teams. The bodyguards are the members whose tasks can be listed as protecting the principal by shielding him or her, diverting potential targets, rescuing the principal from immediate vicinity of an incident, and neutralizing attacker by means of combat techniques. The bodyguards take necessary actions regarding to the aforementioned tasks while they are escorting the principal who goes from one place to another either in a vehicle or on foot. The PES team comprise of individuals with different roles depending on the requirement and threat assessment of the client. These roles include team leader, bodyguard, point man, left flank, right flank, tail, and rear.

Bodyguard Positioning

Before describing the two algorithms for bodyguard positioning, let us first describe the assessments of threats for different scenarios. In order to assess the risks in a particular configuration, we define the metric *ThreatLevel*. *ThreatLevel* is a normalized metric, having values between 0 and 1 inclusive. For a given configuration of a scenario, 0 means that the environment is a safe heaven while 1 means that the environment is a death trap for the VIP. A value in-between suggests the intensity of the physical harm from civilians to the VIP. The assessment of the threat level can either be done by the robotic bodyguards or a human operator with the perspective of a third person.

There are various parameters involved in the assessment of *ThreatLevel*. These parameters are associated with person's perception of the environment. We consider the perception of robotic bodyguard. The major goal of the robotic bodyguard is to have 360-degree situation awareness and the ability to observe and describe a suspect in a useful amount of details. This assessment for immediate response in the unknown crowd depends on various physical and physiological attributes learned by human operatives over the period of training and experience in real life. We are performing *ThreatLevel* assessment according to distances of the civilians in the crowd from the VIP.

In the scenario we are considering that a very important person (VIP) is protected by a group of bodyguard robots $R = \{r_1, r_2, \dots, r_p\}$. In the environment, we have a number of *civilians* $G = \{g_1, g_2, \dots, g_k\}$, each of which can be considered as a potential threat. The robotic bodyguards aim to minimize the risk of physical harm to the VIP by providing effective physical cover at any moment by placing themselves in the strategically optimal positions and prevent attackers to reach their target.

We first propose an approach for the single robotic bodyguard positioning based on *Threat Vector Resolution* (TVR) algorithm. Later, we expand our approach considering multiple robot bodyguards using *Quadrant Load Balancing* (QLB) algorithm. We assume that all agents in the environment are moving in two dimensional space. In other words, we do not consider aerial attacks or threat appearances from higher elevations in our approach. Moreover, in two dimensional space we assume that agents cannot see each other when there exist an obstacle, agent or a human in their line of sight (LoS).

Definition 1. *The function $LoS(x,y) \in \{0,1\}$ specifies whether the agent x can observe the other agent y or not. This function is commutative.*

Definition 1 provides the condition for the two agents to observe each other, i.e. as long as there is no obstacle in between them they can see each other. LoS is dependent on the vision capability of the agent. An assumption has been made here that observer can see the object at the horizon if there is no obstacle in between.

The probability of successful protection of the VIP depends on the position of the robotic bodyguard. We define the circle with radius r_r as the protection circle of a robotic bodyguard r. Each robotic bodyguard can prevent an attack on VIP within its protection circle. Therefore, all robot bodyguards whose protection circle can cover VIP are able protect him. As the distance outside protection circle increases, strength of the protection defined as D_r decreases.

Definition 2. *Let $D_G \in [0,1]$ be the probability that a set of bodyguards can intercept the threat. Then $D_{G_1} \leq D_{G_2}$ if $|G_1| \leq |G_2|$ i.e. more bodyguards provide better protection. D_G given at time instant t as*

$$D_G(t) = \prod_w D(G_w, t) \tag{1}$$

Definition 3. *The maximum threat distance, $MaxDist$, is defined such that if $Dist(VIP, g_i) \geq MaxDist$ then crowd member g_i can not harm the VIP.*

The maximum threat distance assumes that the robotic bodyguard would be able to provide safety cover to the VIP from attacker if it starts outside the $MaxDist$. When moving through the crowd, any civilian closer than $MaxDist$ have certain probability of harming the VIP that will be considered for the threat level metric.

$$ThreatLevel(g_i, VIP) = \begin{cases} LoS(g_i, VIP) \cdot \Phi \cdot \mathrm{e}^{-M\cdot(Dist(g_i,VIP))} \\ \quad \text{if } Dist(g_i, VIP) < MaxDist \\ 0 \quad \text{if } Dist(g_i, VIP) \geq MaxDist \end{cases} \tag{2}$$

where $Dist(g_i, VIP)$ is the distance between the civilian and VIP. Φ and M are the constants which define the slope and the magnitude of the risk curve respectively.

Threat Vector Resolution The goal of TVR is to find the best possible location for a robotic bodyguard to shield VIP against those civilian who have the highest threats value in the configuration. Each robotic bodyguard performs this computation independently without any communication with another robotic bodyguard. We made the assumption that all robotic bodyguards are aware of each others location. In Algorithm 1, TVR takes the set of civilians $\mathbb{S} = \{S_1, ..., S_n\}$ as candidates to evaluate the best available location for a robotic bodyguard to position. The weight of the vector is decided by α which defines the chance of attack by

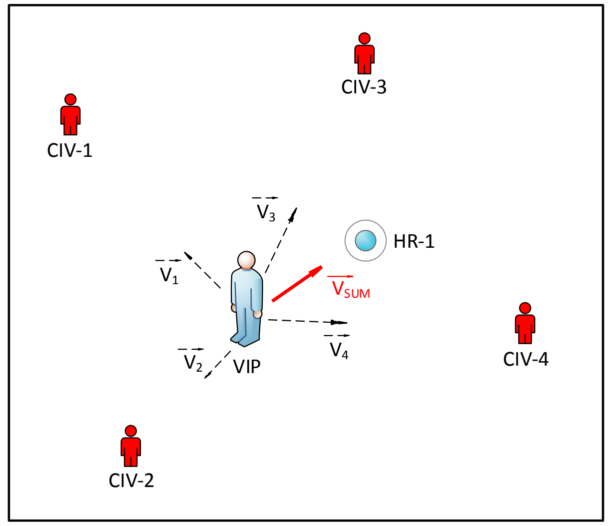

Figure 2: Positioning of a single robot with TVR.

an agent S_i. α is calculated using *ThreatLevel* metric. P_{VIP} represents the current location of the VIP. The resultant vector for avoiding maximum threat in the configuration which provides the suitable location can be found by equation:

$$P_r \leftarrow P_{VIP} + \vec{V} \cdot MaxDist, \tag{3}$$

where P_r is the new location of the single robotic bodyguard r.

Fig. 2 illustrates the use of TVR for an example configuration. In this configuration, four civilian located at different distances from VIP and a single robotic bodyguard. CIV-1 and CIV-2 are closer to VIP compared to the other two civilians. However, the threat vectors $\vec{V_1}$ and $\vec{V_2}$ have smaller magnitudes, since the threat assessments are based on the *ThreatLevel* metric. The robotic bodyguard HR-1 computes the sum vector $\vec{V}_{sum}$ which is directed between $\vec{V_3}$ and $\vec{V_4}$. $\vec{V}_{sum}$ is used for the positioning of the robot.

Algorithm 1 Threat Vector Resolution (TVR)

procedure ThreatVectorResolution($\mathbb{S}$, P_{VIP})
  $\vec{V} = (0,0)$
  $\alpha_i = 0$
  for all $S_i \in \mathbb{S}$ **do**
    if $LoS(S_i, P_{VIP}) == 1$ **then**
      $\alpha_i \leftarrow ThreatLevel(S_i, P_{VIP})$
      $\vec{V} \leftarrow \vec{V} + \frac{\vec{V}(S_i, P_{VIP})}{||\vec{V}(S_i, P_{VIP})||} \cdot \alpha_i$
    else
      continue
    end if
  end for
  return Threat vector $\vec{V}$
end procedure

Quadrant Load Balancing We propose an approach, which we named *Quadrant Load Balancing* (QLB), for bal-

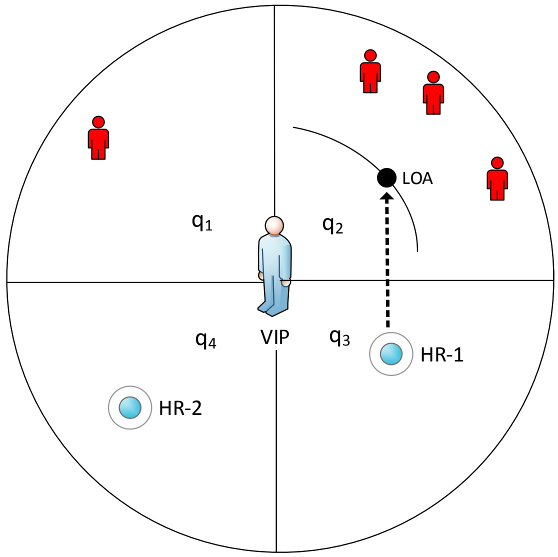

Figure 3: Assigning the closest robot to the quadrant q_2 with highest load.

ancing the load of the threats from civilians by deploying multiple robotic bodyguards. In this approach, the protection circle of a VIP is divided into quadrants and the load of the quadrants shared among the robotic bodyguards. The computation of load of the quadrants is based on the threat vectors, as shown in Algorithm 1 given S as the set of civilians and P_{VIP} as the current location of VIP.

Algorithm 2 handles the assignment of the robotic bodyguards to the quadrants and load balancing. In this algorithm, $Q = \{q_1, q_2, q_3, q_4\}$ is the set of quadrants, where S_{q_i} is the set of civilians in respective quadrant q_i. $R = \{r_1, r_2, \ldots, r_k\}$ represents the set of unassigned robots, where k is initially equal to the total number of robotic bodyguards. $\vec{L}_q$ is the vector sum of the threat vectors of the civilians in a quadrant (S_q). The algorithm iteratively assigns robotic bodyguards and updates the loads of the quadrants after each assignment. $Workload$ function represents the workload which can be handled by a single robotic bodyguard r and $Dist(a, b)$ represents the distance between two points.

Definition 4. *For a given quadrant q and any robot r, $LOA(\vec{L}_q) = P_r$ specifies the location of the robot $P_r = (x, y)$ on the quadrant q.*

$LOA(\vec{L}_q)$ is the *Location on Arc Function* which is estimated for the quadrant q. The current load $\vec{L}_q$ is the sum of the threats in the quadrant computed by Algorithm 1. LOA defines the location at which the robotic bodyguard is placed as follows.

$$LOA(\vec{L}_q) = \begin{cases} P_{VIP} + \vec{L}_q \cdot MaxDist & \text{if unoccupied} \\ LOA(\vec{L}'_q) & \text{otherwise} \end{cases} \tag{4}$$

where

$$\vec{L}'_q = \big(\vec{L}_q.x \cdot cos(\theta), \vec{L}_q.y \cdot sin(\theta)\big), \tag{5}$$

such that θ is the minimum angle which produces an unoccupied location in any one of the two directions for $LOA(\vec{L}'_q)$. We define a location as occupied if there exists another robotic bodyguard or an obstacle on it. Hence, LOA function produces an unoccupied location having the same distance from P_{VIP} and being closest to the ideal case of $P_{VIP} + \vec{L}_q \cdot MaxDist$.

At each iteration of the while loop in Algorithm 2, the quadrant which currently has the most load (q_{max}) is assigned to one of the unassigned robotic bodyguards. The selected robotic bodyguard has the minimum distance ($r_{closest}$) to the location which the robot will take place on the quadrant if it is assigned, while this location in the quadrant is found by LOA, which provides closest available position. After the robot $r_{closest}$ is assigned to q_{max}, it is removed from the set of unassigned robotic bodyguards and the load of the quadrant $L_{q_{max}}$ is decreased according to the $Workload(r_{closest})$.

Fig. 3 illustrates an example case of QLB. In this figure, the quadrants q_3 and q_4 have no civilians ($L_{q_3} = L_{q_4} = (0, 0)$) and $q_{max} = q_1$. Among the two unassigned robot bodyguards, HR-1 is assigned due to its closeness to the LOA. In the second iteration, the load of q_1 will be updated and HR-2 will be assigned to the quadrant with maximum load, among the loads L_{q_2} and L_{q_1} (the remaining load of q_1).

Simulation Study

Simulation Setup

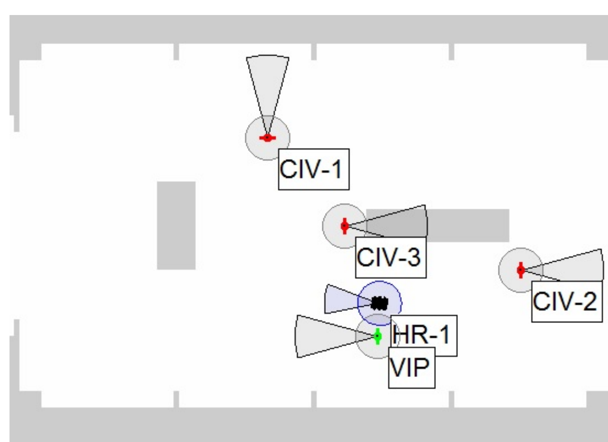

Figure 4: Security scenario displaying robotic bodyguard as HR-1 protecting VIP. CIV-1,2,3 stand for civilians in the simulation.

Algorithm 2 Quadrant Load Balancing (QLB)

procedure QUADRANTLOADBALANCING($\mathbb{S}$, P_{VIP})
  $Q = \{q_1, q_2, q_3, q_4\}$
  $\mathbb{S} = S_{q_1} \cup S_{q_2} \cup S_{q_3} \cup S_{q_4}$
  $R = \{r_1, r_2, \ldots, r_k\}$
  for all $q \in Q$ **do**
    $\vec{L}_q \leftarrow ThreatVectorResolution(S_q, P_{VIP})$
    $L \leftarrow L \cup \vec{L}_q$
  end for
  $c = 0$
  while $c < k$ **do**
    $q_{max} \leftarrow null$
    $L_{q_{max}} = 0$
    for all $\vec{L}_q \in L$ **do**
      if $\left|\left|\vec{L}_q\right|\right| > ||L_{q_{max}}||$ **then**
        $L_{q_{max}} \leftarrow \vec{L}_q$
        $q_{max} \leftarrow q$
      end if
    end for
    $r_{closest} \leftarrow null$
    $Dist_{min} = \infty$
    for all $r \in R$ **do**
      if $Dist(r, q_{max}) < Dist_{min}$ **then**
        $Dist_{min} = Dist\big(r, LOA(\vec{L}_{q_{max}})\big)$
        $r_{closest} \leftarrow r$
      end if
    end for
    $P_{r_{closest}}(x, y) \leftarrow LOA(\vec{L}_{q_{max}})$
    $R \leftarrow R - r_{closest}$
    if $Workload(r_{closest}) < ||L_{q_{max}}||$ **then**
      $\vec{L}_{q_{max}} \leftarrow \vec{L}_{q_{max}} \cdot \frac{||L_{q_{max}}|| - Workload(r_{closest})}{||L_{q_{max}}||}$
    else
      $\vec{L}_{q_{max}} \leftarrow (0, 0)$
    end if
    $c = c + 1$
  end while
end procedure

We carried out simulation experiments using Yaes Simulator (Bölöni and Turgut 2005), which is developed by our research group. Moreover, three dimensional visualizations of the scenarios are created using V-REP 3D simulator (Freese et al. 2010) as shown in Fig. 1.

Fig. 4 shows the simulation of providing security to VIP by single robotic bodyguard, whereas Fig. 5 shows the simulation of multiple robotic bodyguards protecting VIP. Gray colored regions in the simulation represents the obstacles and the boundaries (walls). Civilians (threats) are represented with prefix CIV-. VIP is surrounded by autonomous humanoid robotic bodyguards with prefix HR-. Gray circle represents the personal zone of the agents inside which attack by a civilian can be considered undesirable. Gray outward cone represents the movement and vision direction of the agent. All the civilians and VIP in the simulation can be manually controlled using keyboard or game controller by human testers while robotic bodyguard agents are autonomous. The simulation can be controlled in two modes: Run the simulation by controlling an agent one step at a time or the simulation continues for predefined configuration. These controls provide us ability to control and evaluate different configurations as well as possible sudden changes in configurations.

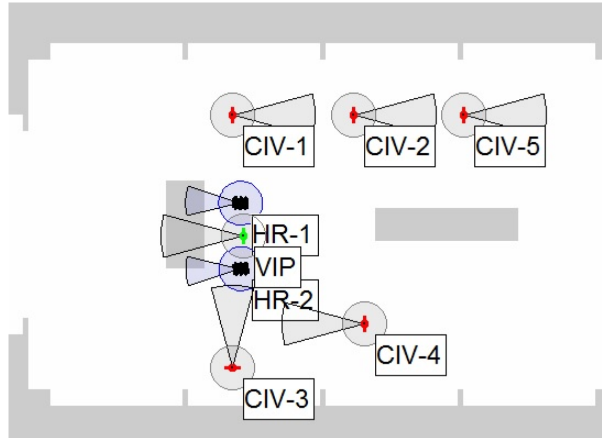

Figure 5: Security scenario displaying robotic bodyguards as HR-1, HR-2 protecting VIP. CIV-1,2,3,4,5 stand for civilians in the simulation.

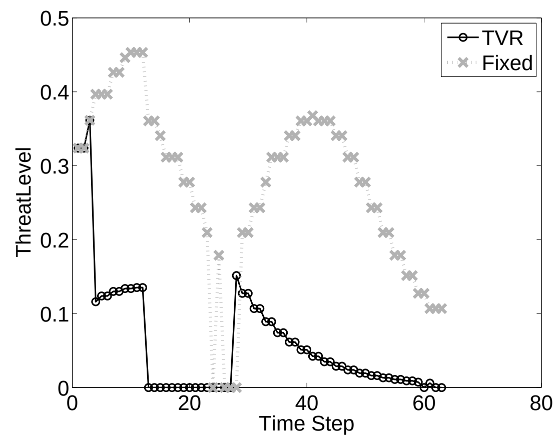

Figure 6: *ThreatLevel* comparison of fixed robotic bodyguard orientation vs TVR in the single robotic bodyguard simulation.

Performance Results

In swarm robotics, algorithms focus on maintaining fixed swarm formations while performing path planning. We compared our approaches against the fixed formation in order to assess decrease in *ThreatLevel* of the configuration. Fig. 6 shows the *ThreatLevel* comparison of a fixed robotic bodyguard movement (left or right flank), against the TVR approach in a single robot bodyguard simulation. Moreover, Fig. 7 shows the mitigated *ThreatLevel* of

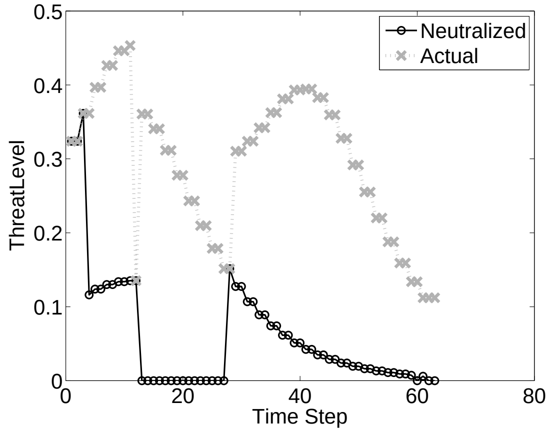

Figure 7: Comparison of Actual *ThreatLevel* vs Neutralized *ThreatLevel* in the single robotic bodyguard simulation.

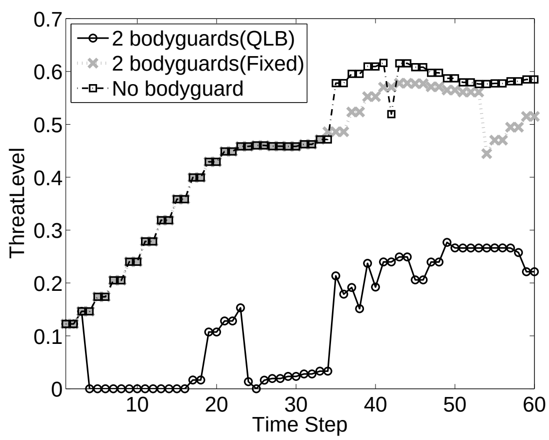

Figure 8: *ThreatLevel* comparison of fixed robotic bodyguard orientation vs QLB in the multiple robotic bodyguards simulation.

the scenario in the absence and presence of single robotic bodyguard. This graph demonstrate the overall reduction in *ThreatLevel* called as ‘Neutralized’ *ThreatLevel* against ‘Actual’ *ThreatLevel* when the VIP is not accompanied by a robotic bodyguard.

For the multiple robot bodyguards scenario, Fig. 8 shows the *ThreatLevel* comparison of fixed multiple robot bodyguards movement (left and right flank), against the QLB approach. In the fixed strategy, both robotic bodyguards acquire either left behind or right behind position and follow VIP while avoiding obstacles during the walk. Fixed strategy is not appropriate for providing security in this problem as shown in the graph. ‘Neutralized’ *ThreatLevel* provided by ‘2 bodyguards (Fixed)’ is as good as ‘No bodyguard’ in the scenario where as ‘2 bodyguards (QLB)’ worked in collaboration to bring down average value of *ThreatLevel* to 1.5 by providing sufficient cover using QLB algorithm.

The comparison of the level of security by variable team

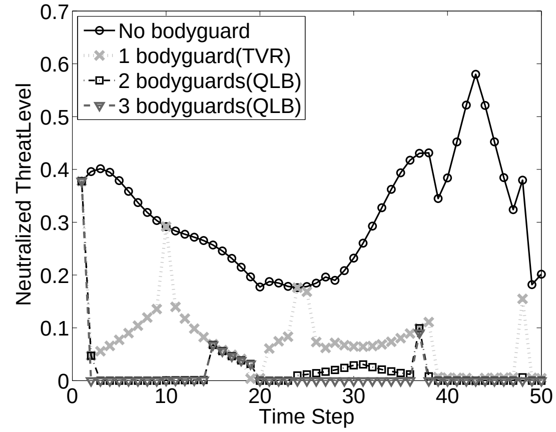

Figure 9: Comparison of Neutralized *ThreatLevel* of No bodyguard, 1 bodyguard running TVR, 2 and 3 bodyguards running QLB.

sizes of robotic bodyguards is made in Fig. 9. This simulation consists of six civilians occupying different locations on the map. ‘1 bodyguard’ manages to provide sufficient body cover using TVR algorithm most of the time, but it is still unable to protect VIP at time steps 10, 25 and 48 where it equates to Neutralized *ThreatLevel* of ‘No bodyguard’. Multiple robotic bodyguard teams using QLB algorithm demonstrate very close performance in providing security as shown by ‘2 bodyguards’ and ‘3 bodyguards’. Moreover, more number of bodyguards provides better protection as shown around time step 30 where ‘3 bodyguards’ provides better protection to VIP compared to ‘2 bodyguards’ for the same instants.

Conclusions

In this paper, we focused on the positioning of multiple robotic bodyguards during their movements to protect VIP in the scenario. We proposed *Threat Vector Resolution* approach for single robot bodyguard positioning and *Quadrant Load Balancing* for collaborative security using multiple robot bodyguards. We evaluated the proposed approaches against rigid formation of robotic bodyguards by the simulation experiments.

As a future work, we intend to expand our simulation to include more realistic scenarios for the current behavior. Moreover, we want to consider explicit communication among robotic bodyguards as well as among human and robotic bodyguards. This feature can be implemented using an auction-based bidding approach with respect to current scenario. While the proposed approaches in this paper are formulated by the definitions, we plan to expand them with automated learning algorithms for real behavior learning based on imitation.

Acknowledgments. The research reported in this document/presentation was performed in connection with Contract Number W911NF-10-2-0016 with the U.S. Army Research Laboratory. The views and conclusions contained in this document/presentation are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of the U.S. Army Research Laboratory, or the U.S. Government unless so designated by other authorized documents. Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Agmon, N.; Kaminka, G. A.; and Kraus, S. 2011. Multi-robot adversarial patrolling: Facing a full-knowledge opponent. *Journal of Artificial Intelligence Research* 42:887–916.

Bölöni, L., and Turgut, D. 2005. YAES - a modular simulator for mobile networks. In *Proc. of the 8-th ACM/IEEE Int'l Symp. on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2005)*, 169–173.

Cao, Y. U.; Fukunaga, A. S.; and Kahng, A. 1997. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots* 4:7–27.

Fazli, P.; Davoodi, A.; and Mackworth, A. 2013. Multi-robot repeated area coverage. *Autonomous Robots* 34:251–276.

Freese, M.; Singh, S.; Ozaki, F.; and Matsuhira, N. 2010. Virtual robot experimentation platform v-rep: A versatile 3d robot simulator. In *Simulation, Modeling, and Programming for Autonomous Robots*, volume 6472. 51–62.

Khan, S. A.; Arif, S.; and Bölöni, L. 2014. Towards learning movement in dense crowds for a socially-aware mobile robot. In *Workshop on Adaptive Learning Agents (ALA-2014)*.

Khan, S. A.; Bhatia, T. S.; Parker, S.; and Bölöni, L. 2012. Modeling human-robot interaction for a market patrol task. In *Proc. of 25th International FLAIRS Conference*, 50–55.

Maurette, M. 2003. Mars rover autonomous navigation. *Autonomous Robots* 14:199–208.

Pita, J.; Jain, M.; Marecki, J.; Ordez, F.; Portway, C.; Tambe, M.; Western, C.; Paruchuri, P.; and Kraus, S. 2008. Deployed ARMOR protection: the application of a game theoretic model for security at the Los Angeles International Airport. In *Proc. of the 7th Int'l Conf. on Autonomous Agents and Multiagent Systems*, 125–132.

Pita, J.; Tambe, M.; Kiekintveld, C.; Cullen, S.; and Steigerwald, E. 2011. GUARDS: Innovative application of game theory for national airport security. In *IJCAI Proc. of Int'l Conf. on Artificial Intelligence*, volume 22, 2710.

Portugal, D., and Rocha, R. P. 2013. Distributed multi-robot patrol: A scalable and fault-tolerant framework. *Robotics and Autonomous Systems* 61(12):1572 – 1587.

Raibert, M.; Blankespoor, K.; Nelson, G.; Playter, R.; et al. 2008. Bigdog, the rough-terrain quadruped robot. In *Proc. of the 17th World Congress*, 10823–10825.

Schneider, G. 2009. *Beyond the Bodyguard: Proven Tactics and Dynamic Strategies for Protective Practices Success*. Universal-Publishers.

Shieh, E.; An, B.; Yang, R.; Tambe, M.; Baldwin, C.; DiRenzo, J.; Maule, B.; and Meyer, G. 2012. PROTECT: A deployed game theoretic system to protect the ports of the united states. In *Proc. of the 11th Int'l Conf. on Autonomous Agents and Multiagent Systems-Vol. 1*, 13–20.

Solmaz, G., and Turgut, D. 2013. Event coverage in theme parks using wireless sensor networks with mobile sinks. In *Proceedings of IEEE ICC'13*, 1522–1526.

Solmaz, G., and Turgut, D. 2014. Optimizing event coverage in theme parks. *Wireless Networks (WINET) Journal* 20(6):1445–1459.

Tsai, J.; Kiekintveld, C.; Ordonez, F.; Tambe, M.; and Rathi, S. 2009. IRIS - a tool for strategic security allocation in transportation networks. In *Proc. of 8th Int'l Conf. on Autonomous Agents and Multiagent Systems*, 37–44.

Vanek, O.; Bosansky, B.; Jakob, M.; and Pechoucek, M. 2010. Transiting areas patrolled by a mobile adversary. In *IEEE Symposium on Computational Intelligence and Games (CIG)*, 9–16.

Varakantham, P.; Lau, H. C.; and Yuan, Z. 2013. Scalable randomized patrolling for securing rapid transit networks. In *Proc. of the 25th Innovative Applications of Artificial Intelligence Conf. (IAAI)*, 1563–1568.

Yin, Z.; Jiang, A. X.; Tambe, M.; Kiekintveld, C.; Leyton-Brown, K.; Sandholm, T.; and Sullivan, J. P. 2012. TRUSTS: Scheduling randomized patrols for fare inspection in transit systems using game theory. *AI Magazine* 33(4):59.