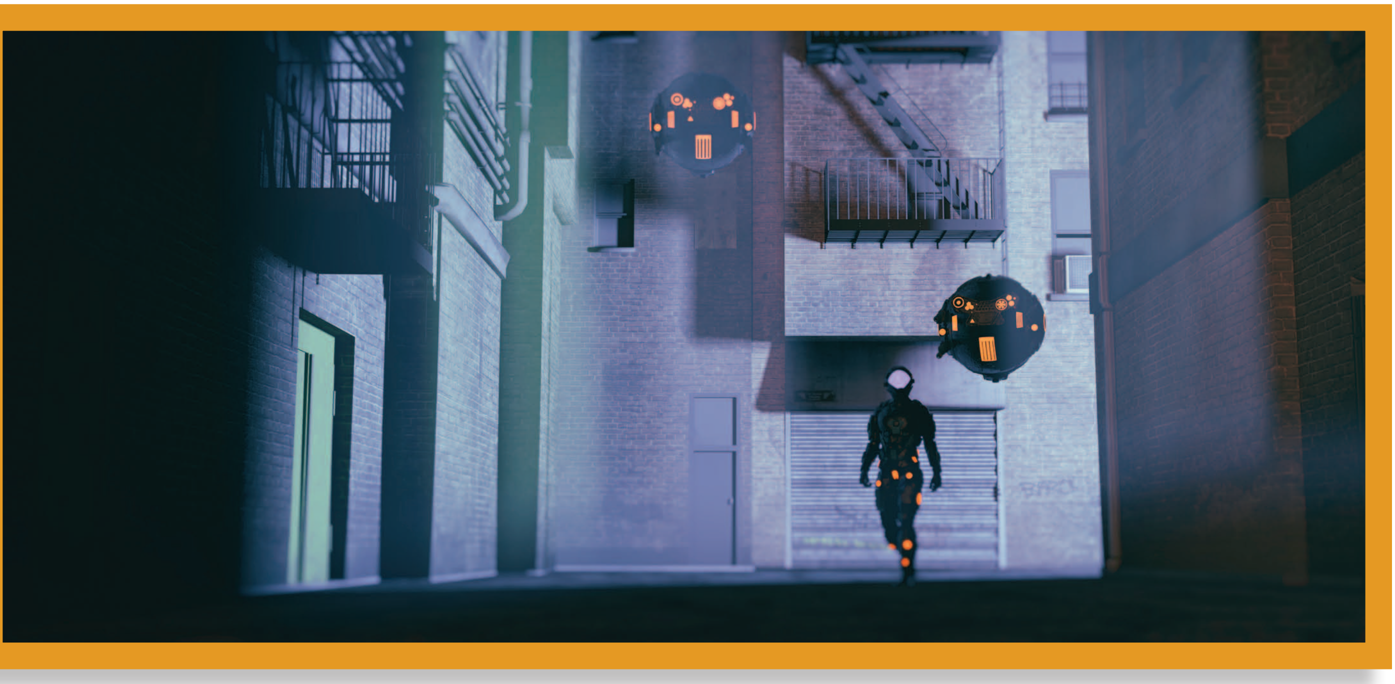
*A Decision-Making Framework for Networked Multiagent Systems*



©ISTOCKPHOTO.COM/ MATJAZ SLANIC

# Autonomous Surveillance Robots

By Stefan Witwicki, José Carlos Castillo, João Messias, Jesús Capitán, Francisco S. Melo, Pedro U. Lima, and Manuela Veloso

This article proposes an architecture for an intelligent surveillance system, where the aim is to mitigate the burden on humans in conventional surveillance systems by incorporating intelligent interfaces, computer vision, and autonomous mobile robots. Central to the intelligent surveillance system is the application of research into planning and decision making in this novel context. In this article, we describe the robot surveillance decision problem and explain how the integration of components in our system supports fully automated decision making. Several concrete scenarios deployed in real surveillance environments exemplify both the flexibility of our system to experiment with different representations and algorithms and the portability of our system into a variety of problem contexts. Moreover, these scenarios demonstrate how planning enables robots to effectively balance surveillance objectives, autonomously performing the job of human patrols and responders.

## Toward Intelligent Decision Making

Combining recent research advances in computer vision, robot autonomy, and artificial intelligence (AI) has the potential to revolutionize surveillance technology. Consider the careful attention spent by security personnel to monitor numerous live video feeds from cameras that are presently

*Digital Object Identifier 10.1109/MRA.2017.2662222*
*Date of publication: 4 August 2017*

observing our parking lots, university campuses, and shopping malls. Imagine the monotonous patrols of armies of security guards through countless corridors. Deliberate over the difficult strategic decisions of where and how to allocate precious human resources, both in response to immediate security concerns and in anticipation of future conditions. To maintain safety and security, the conventional surveillance system relies critically on human attention, action, and intelligence. However, such reliance is untenable in a society where the trend is toward more cameras embedded in larger and more complex environments to defend against a growing array of potential threats (from burglary to natural disasters to terrorist attacks). Here, we advocate a shift toward reliance on autonomous system components so that society may scale up to meet present-day surveillance needs.

One aspect of this topic that has received considerable attention from researchers is real-time scene analysis. Systems have already been developed to autonomously analyze video streams in environments such as transportation networks [6], [27] and public spaces [5] to identify actors and characterize their behavior. Recent examples include IBM's Smart Surveillance System project [22] and Yao et al.'s system for cooperative object tracking [30]. There are also approaches for activity interpretation [8], [12], [13], [20], [25], while other works are more focused on meeting low-bandwidth requirements by locally processing video-feed images [4]. Although these systems can autonomously extract relevant information for surveillance purposes, they are still heavily dependent on a team of human security personnel to perform such actions as covering areas that may be outside the range of the stationary sensor network and resolving situations that may require physical intervention. Our work aims to increase autonomy and to reduce the human burden by introducing autonomous mobile robots into the system.

Research in robot mobility has advanced to the point that robots now have the ability to navigate complex environments, patrolling as human guards would. Equipped with cameras and other sensors, they can also serve as mobile surveillance nodes to augment a network of statically situated cameras. For instance, a robot can provide temporary coverage of areas that may become critical because of camera failures or other anomalies. Moreover, robots have the mobility, sensors, and actuators to respond directly to events detected over fixed camera streams, thereby leveraging real-time scene analysis.

To integrate these complementary research technologies effectively and to render robots truly autonomous requires a third key technology: intelligent decision making. Robots should choose their actions so as to fulfill a combination of objectives, given limited resources. This is often framed as a robot task selection and allocation problem [10] and has been approached through a variety of AI techniques, from logic-based (classical) planning methods [9], to market-based (auction) solutions [15] and those relying on constraint optimization [16]. An obstacle to applying such techniques here is that surveillance decisions are riddled with uncertainty. Uncertainty is present in robots' awareness because of imperfect sensing and localization as well as in environmental dynamics because of imprecise control and unpredictability about when security events may occur. This challenge leads us to examine state-of-the-art formalisms for modeling robots' task dynamics and for planning under uncertainty—formalisms that push the boundaries of robot autonomy.

The primary contribution of this work, however, is the integration of complementary research technologies from video surveillance, mobile robotics, and AI. We demonstrate the efficacy of our integration through a prototype system that includes a small number of robots and cameras deployed in realistic environments. A modular architecture and general framework for representing and communicating surveillance events makes our system a useful test bed for experimenting with various research technologies. In contrast to past results in multirobot patrolling that employed human operators to orchestrate the robots' behavior [7], we are able to achieve fully autonomous security robots capable of making decisions on their own, with the potential to help human operators.

## Overview of the Surveillance Framework

We begin with a brief overview of our framework, which is motivated by a concrete example of a decision faced by a patrolling robot. This leads us to characterize the decision-making problem as well as to structure our system in support of the implementation and testing of decision-theoretic planning for mobile surveillance robots.

### Motivating Example

Imagine adding a robot to the observation environment shown in Figure 1. In contrast to the static cameras placed at fixed positions, the robot is capable of dynamically patrolling the building. It can move from room to room, using its

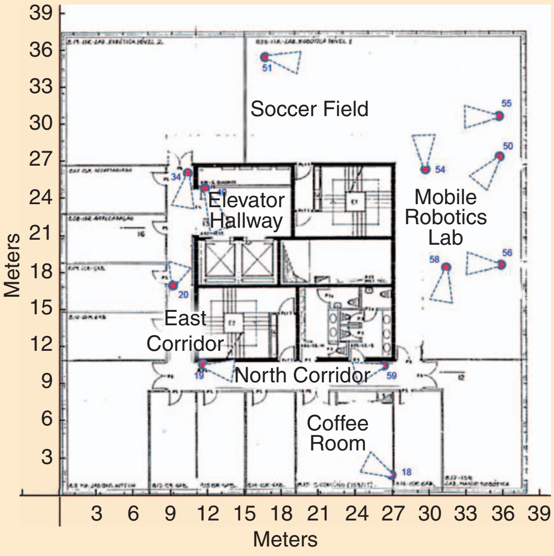


**Figure 1.** A staged indoor surveillance environment with the positions of the static cameras (red circles) and the common coordinate system for event location.

sensors to scan for anomalies that the static cameras might have missed and using its actuators to interact with the environment in ways that a static camera cannot. The robot's limitation, however, is that it can occupy only one physical location at a time.

Consider that, late one night, the robot is patrolling the east corridor on its way to the elevator hallway. Suddenly, one of the fixed cameras detects a person moving in the north corridor. At this time of day, the north corridor has restricted access, arousing suspicion that someone is trespassing. Assuming this event is communicated to the robot across the network, the robot could turn around and proceed directly to that location. Alternatively, the robot could continue along to inspect the elevator hallway, which is also an important room in the building. This example illustrates the kind of relevant decisions that a patrolling robot could face, given its current status and the status of the surveillance system. The decision whether to respond immediately to an event or to continue patrolling should be made carefully and deliberately, since it could compromise the security of the building.

**Table 1. The challenges of surveillance decision making.**

| Challenge | Explanation |
|---|---|
| Constrained resources | A robot has a finite operation time and cannot visit all locations instantaneously. |
| Urgency/priority | A trespassing event left unaddressed for too long can turn into a robbery. |
| Uncertainty about event occurrences | It is unknown when, where, and even if an event will occur. |
| Uncertainty in decision consequences | There is no guarantee that the robot will succeed in its actions, e.g., thwarting the trespasser. |
| Uncertainty in the sensor data | Imperfect detection methods may yield false positives and false negatives. |
| Coordination of decisions | A robot team should handle events in parallel, avoiding redundancy. |
| Intermittent communication | This can occur, for example, when robots traverse large and complex spaces with dead zones. |

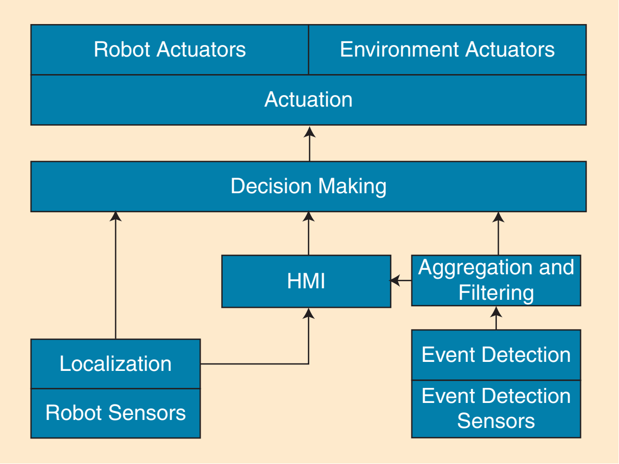


**Figure 2.** The modular design of our surveillance framework.

### Modular System Design for Decision Making

In general, a mobile security robot will experience a sequence of decision making about where to go and what to do, as long as it is operating in the environment and events are being detected by the network. To increase the autonomy of the networked robotic system, planning methodologies should consider several relevant aspects within the decision-making problem, as summarized in Table 1.

In addition to accommodating various decision-making methodologies, an effective autonomous surveillance framework needs to deal with a wide range of heterogeneous sensors and actuators exchanging information in real time, e.g., differing robot platforms, lasers, cameras, microphones, and speakers. Therefore, we propose a modular framework for security inspection that divides the overall system into components and defines a set of interfaces for component interaction and communication. The system is versatile enough to allow for adaptable reuse as well as the incorporation of new functionalities (e.g., new sensor technologies).

Figure 2 diagrams our modular surveillance framework. Apart from the heterogeneous sensor and actuator modules, a human–machine interaction (HMI) module is included to display information (e.g., detected events) to the operator, to receive remote commands (e.g., sending a robot to a desired position), and to produce audible signals from each robot in the form of speech, whereby the robot can interact with people in the environment.

## Detecting and Disseminating Events

Events, such as a person requiring assistance or an intrusion, form the basis for all intelligent surveillance activities. In this section, we describe where these events come from and how they are automatically detected and represented in support of effective robot planning. For illustrative purposes, we focus our description on the trespassing event introduced in the "Motivating Example" section.

### Image Processing

The multicamera system requires live video acquisition and transmission. High-resolution camera images need to be captured and received at a steady rate and reliably enough to perform event detection. This involves high-bandwidth computation, balanced across several high-performance servers, each processing the images in real time.

Our surveillance system integrates the technique proposed in [19] for both detecting people as they move around within parameters of the designated area and for sensing other events, such as a request for assistance or trespassing. Other image-processing algorithms could be plugged in to our system since the framework is flexible, requiring only that new modules respect the interfaces to communicate with connected modules. The processing is divided into two main phases.

1) Human presence is detected by a background subtraction-based algorithm; the person is subsequently tracked via data association between consecutive frames.
2) Human activity is detected by means of a classifier that analyzes a tracked person's movements through optical flow computation.

Table 2 shows the performance of our algorithm for detection of waving compared to some state-of-the-art techniques on the KTH action database (http://www.nada.kth.se/cvap/actions/). In the method employed, the temporal support of the classification of every sequence uses an event window size of 4 s (at 25 frames/s), and considers it a waving event if at least 60% of the single-frame classifications are positive in that sequence. More details and results of our method can be found in [19].

**Table 2. The accuracy of state-of-the-art methods for waving detection.**

| Technique | Accuracy |
|---|---|
| Our method | 91.7% |
| Niebles et al. [20] | 93% |
| Ke et al. (1) [13] | 88% |
| Ke et al. (2) [12] | 91.7% |
| Schüldt et al. [25] | 73.6% |

Continuing with our running example, Figure 3(a) highlights two cameras with overlapping fields of view in the area labeled *North Corridor*. Figure 3(b) illustrates how detection of a person on the image plane is translated into positions on the global coordinate frame of the scenario (depicted using the Figure 1 axes). This coordinate system is shared by all robots, and image coordinates can be translated to it by means of homography-based transformations. Along with the detected positions, we model uncertainties that capture the detection imprecision of the sensor itself [illustrated as ellipses in Figure 3(d)]. False

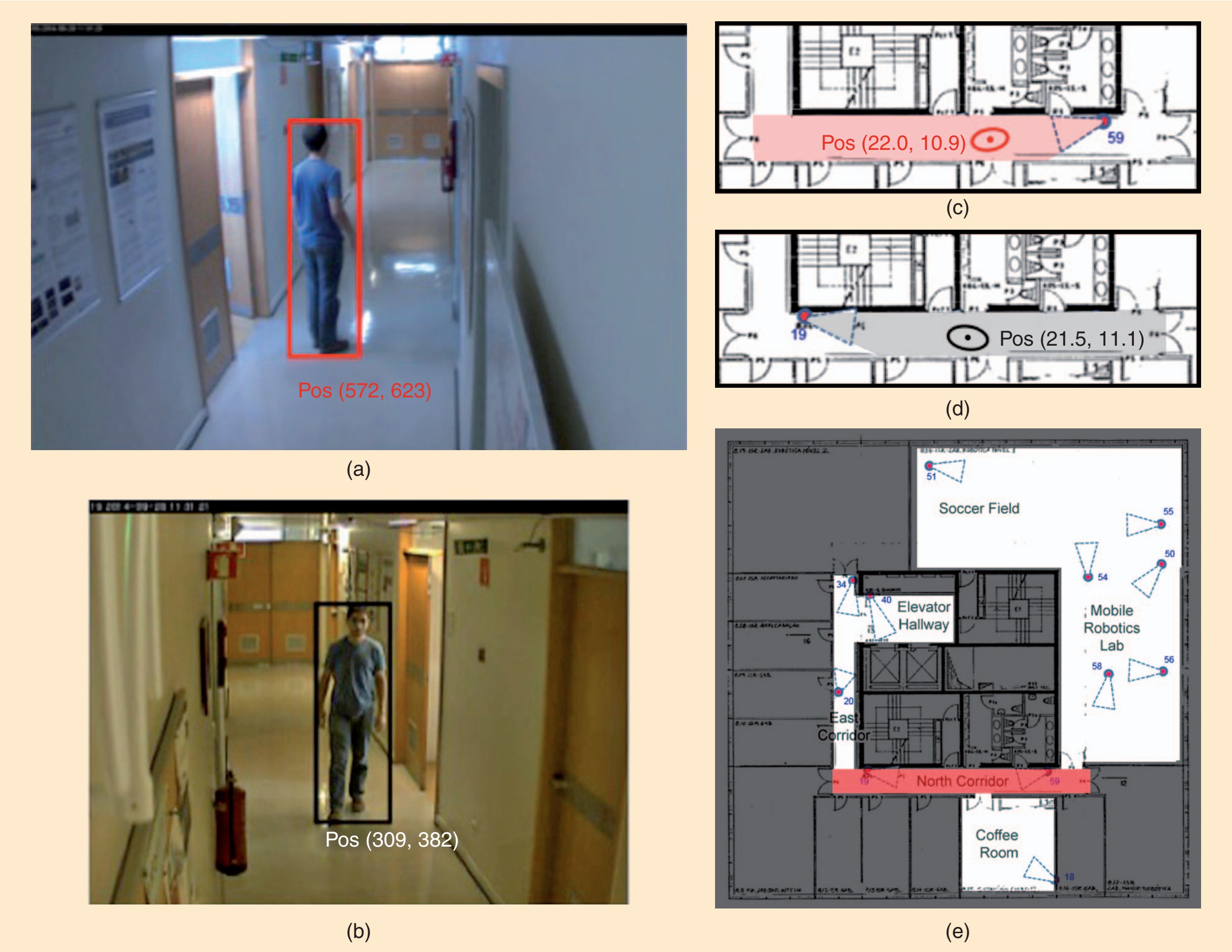


**Figure 3.** Our running example of a trespassing event detection: images acquired by (a) camera 59 and (b) camera 19, with detections; (c) field of view and detection of (d) camera 59 and field of view and detection of camera 19; (e) a scenario abstraction map, where the red zone corresponds to a restricted area, the white zones to accessible areas, and the dark zones to areas unsuitable for robot event attendance (e.g., cluttered zones).

positives (where the detected event did not actually occur) and false negatives (where an event was missed) are thereby modeled probabilistically. Once detected, the events are sent to the Aggregation and Filtering block, as illustrated in Figure 2.

### Event Aggregation and Filtering

To mitigate the noisy measurements produced by state-of-the-art image-processing algorithms and to improve the consistency of human detection, we aggregate information from multiple overlapping cameras. In our system, cameras provide events as three-dimensional positions and orientations with their associated uncertainties (modeled as a 6 × 6 covariance matrix), which are then aggregated together in a centralized fashion. We keep track of the position of every event detected, and once new samples of camera feedback are received, data association is used to match detections of previously identified actors or to distinguish new actors. Data association in our multicamera scenario is solved by methods such as Kullback–Leibler divergence [14].

Figure 3 shows how overlapping cameras can capture the same person and will need the multiple channels of feedback to be combined by the aggregation module. The aggregation module receives samples of feedback asynchronously from multiple cameras and updates the information of the corresponding tracks accordingly (or creates new tracks when required). The event-filtering system recognizes the tracked detection as trespassing by way of a predefined abstraction of the scenario map wherein some areas are marked as forbidden [Figure 3(c)]. Once it has been noticed that a person is trespassing or that there is other relevant human activity, the system generates and transmits a corresponding metaevent to the mobile robots.

## Autonomous Mobile Robot Responders

To play the part of human security guards, mobile robots should be capable of responding to surveillance events regardless of when or where they may occur. The random nature of such events induces a problem of decision making under uncertainty at various levels of abstraction. The robot team should cooperatively decide which robot, if any, should respond to a new event (task allocation); robots should respond to events in the most efficient manner (task execution); and each robot should routinely decide where to position itself in anticipation of an event (navigation). In this section, we describe how the decision-making problems in our surveillance framework are modeled symbolically, enabling their treatment through automated planning and reasoning mechanisms.

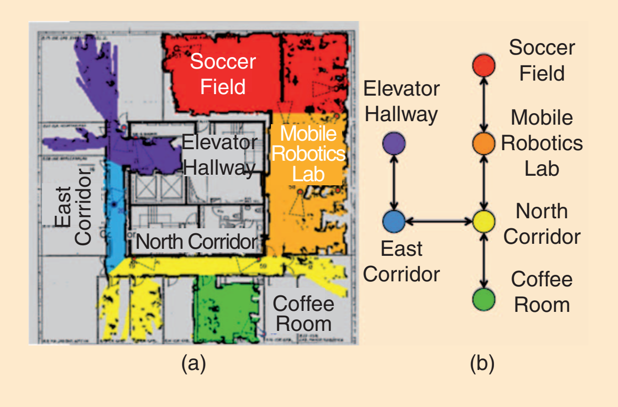


**Figure 4.** (a) A map of the environment of our running example partitioned into areas of interest overlaid with the laser-based map used for robot navigation and (b) the topological graph corresponding to this discretization that is used in the decision-making block of our system.

### Abstracting the System and Its Environment

Consider modeling the autonomous robots' decisions by simulating in detail the many possible detections of events and the various actuations of motors by which each robot could travel to all of the possible event locations. Because of the great many continuous variables involved and the unpredictability of the events, the original optimization problem derived from making low-level decisions may be intractable. To tackle this decision-making problem, it is necessary to describe it at a coarser level of abstraction, including only as much information as that which is deemed relevant to differentiate between the outcomes of the possible decisions of the robots.

First, we partition the environment into a discrete set of locations that can be encoded as a topological graph onto which the position of the robots and the detected events can be mapped. Second, we discretize the space of possible controls for the robots as abstract movement-actions. From each node in the topological graph (describing the location of each robot), there are as many movement actions as adjacent nodes. These actions invoke the robot's lower-level path planner, driving it to a predefined waypoint associated with a graph node (though those actions may fail, leading to nondeterministic transitions). In particular, we assume that the robots are equipped with onboard sensors for localization and navigation. Standard probabilistic localization methods and path-planning algorithms can be used.

The environment of our running example shown in Figure 3, when discretized in the above manner, results in a topological graph describing reachable locations, depicted in Figure 4. This discrete representation of location is then coupled with additional symbolic variables that impact a robot's decisions, for instance, the type and nature of each detected event (e.g., trespassing). The selection of symbolic variables depends on the desired behavior of the system (as we explain in the "Formalizing the Decision-Making Problem" section). Moreover, different automated planning mechanisms may expressly depend on different representations of the environment. For instance, while logic-based planners rely on predicate-based representations of these variables, decision-theoretic planners can operate directly over integer-valued discrete representations. The common thread, however, is a discrete representation of the state of the system as a whole and of the decisions (or actions) that can be performed at the time of each event.

### Formalizing the Decision-Making Problem

Equipped with a symbolic description of the system and of the capabilities of each robot, we can then formalize the decision-making problem. Let $s_t \in \mathcal{S}$ represent the discrete state of the system at some discrete time $t$, which is typically a tuple of symbolic variables. At each time $t$, the robot(s) must select an action $a_t \in \mathcal{A}_t$, where $\mathcal{A}_t$ represents the set of possible symbolic decisions available at that time. The execution of $a_t$ influences the resulting state at the next decision episode $s_{t+1}$.

In our running example, one way of modeling the state is $s_t = \langle r_t, x_t^1, \ldots, x_t^6, b_t \rangle$, where $r_t$ represents the topological position of the robot (within the possible alternatives represented in Figure 4); $x^{1,\ldots,6}$ are the states of each topological node, which could be modeled, for instance, as $x^i \in$ {Unknown, Clear, Intruder}; and $b_t$ represents the battery level of the robot. Additionally, the actions at each time could be the high-level navigation movements between nodes of the topological graph as well as other possible interactions of the robot with its environment, e.g., $\mathcal{A} =$ {Up, Down, Left, Right, ExpelIntruder}.

Given these symbolic representations of states and actions, the general decision-making process can be cast as the following optimization problem: at each time $t$, given the history of states and actions $\langle s_0, a_0, s_1, a_1 \ldots, s_{t-1}, a_{t-1} \rangle$, select a new action $a_t$ to satisfy one of the following optimization targets:

- Maximize a target utility function of future visited states and selected actions (utility-based planning).
- Minimize the number of decisions needed to reach a certain goal state (goal-directed planning).

This formulation of the decision-making process is general enough to encompass most logic-based and decision-theoretic planning methodologies.

### Application of Decision-Theoretic Planners

As suggested in the "Autonomous Mobile Robot Responders" section, decision-theoretic planning methods are especially applicable to the type of problems involved in our multiagent surveillance system since they account for multiple sources of uncertainty in the environment. As such, we have opted to apply these methods to obtain decision-making policies for the robot team in our implementation of the surveillance system.

Most decision-theoretic methods are based on the concept of Markov decision processes (MDPs) or its extensions [3]. An MDP is an instantiation of the decision-making process defined in the "Formalizing the Decision-Making Problem" section, where the state transitions after executing a team action are modeled with a transition probability function and the relative priorities of each state and action (desired behavior) are encoded by a reward function. The objective in an MDP is to obtain a particular mapping of states to actions, $\pi: \mathcal{S} \to \mathcal{A}$ (a policy) that maximizes the expected accumulated reward over a certain (possibly infinite) number of future steps (i.e., utility-based planning).

The definition of the components of an MDP is domain dependent. For instance, in our running example, the transition function depends on the probability that the robot successfully completes its navigation actions and on the probability that an intruder appears in a room. Each time the robot visits a room, its state changes to either Clear or Intruder. If the robot has not visited a room for some time, its state could be reset to unknown, symbolizing a lack of information regarding its occupancy.

Furthermore, a positive reward could be assigned to a state in which all rooms are known to be clear, and, likewise, a negative reward to a room that has an intruder state. No reward would be given for unknown state rooms. Since the robot's policy attempts to maximize reward, this would induce the robot to try to visit all rooms as fast as possible (automatically determining an optimal patrol order), while at the same time prioritizing its response to intruder states. A more specific definition of the transition and reward models for a surveillance task that is analogous to our running example can be found in [29] and in the supplementary material accompanying this article in IEEE *Xplore*.

In some applications, considering the effect of limited or noisy information may be important for decision making. Partially observable MDPs (POMDPs) are an extension of MDPs that also account for uncertainty when observing the state [26], and they are appropriate when the cameras produce unreliable detections. Although calculating policies for POMDPs is computationally more demanding, we demonstrate in the section "Event-Driven POMDPs for Multirobot Surveillance" that this calculation is feasible for a handful of robots, and we discuss in the "Limitations and Extensibility" section how such models could be scaled to larger autonomous surveillance problems.

## Case Studies

In the preceding sections, we have illustrated the various aspects of our autonomous robot surveillance framework using a simple running example. We now turn to several concrete case studies, wherein we formulate and solve the decision-making problem using state-of-the-art planning techniques and deploy the resulting plans in real robots. The case studies involve different environments, events, robot capabilities, and planning algorithms, showcasing the generality of our framework. Specific details on the models used can be found in the supplementary material accompanying this article in IEEE *Xplore*.

### Common Implementation of Components

With the aim of portability and flexibility, we have implemented our surveillance framework, described in the "Overview of the Surveillance Framework" section, on top of the widely adopted Robot Operating System (ROS) infrastructure [24]. Our implementation consists of three macroblocks communicating by means of ROS topics (see Figure 5). First, a robot macroblock is run on each surveillance robot, acting as its intelligence. The modules for robot localization and navigation of our framework described in Figure 2 are here implemented by means of the ROS Navigation Stack, which provides Monte Carlo localization and standard algorithms to

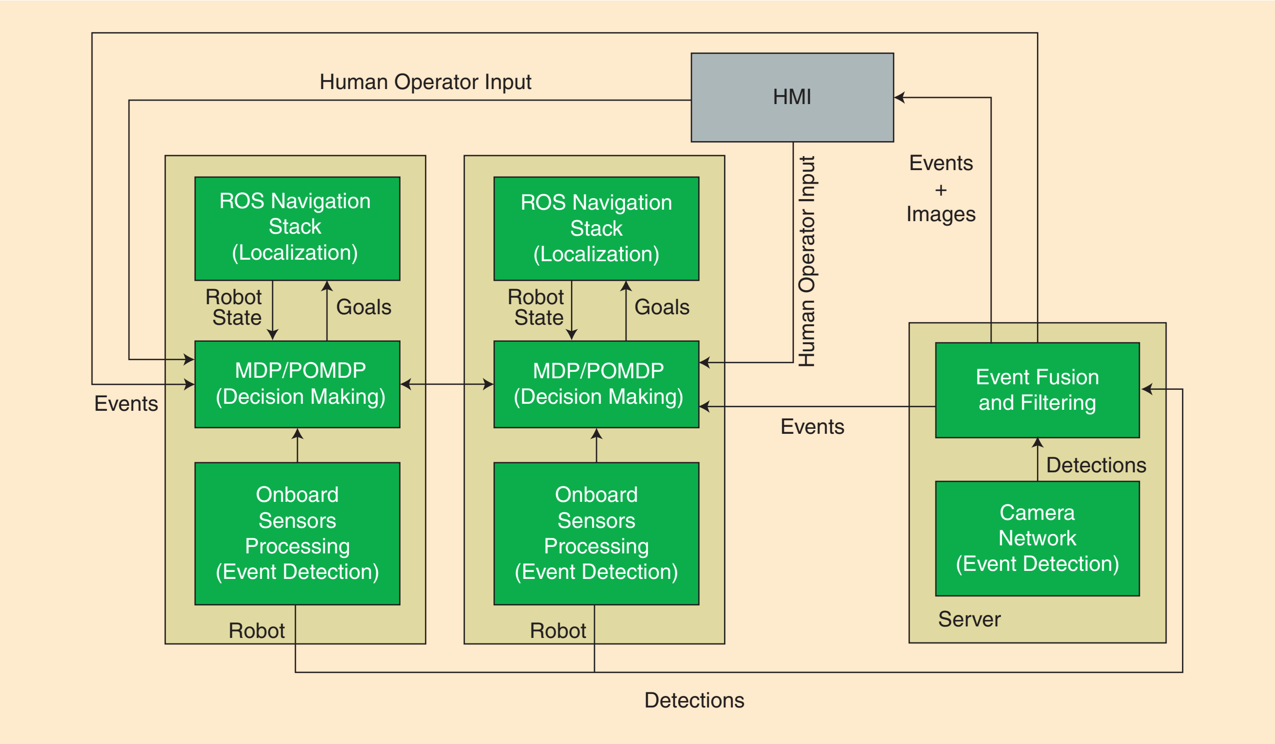

Human Operator Input
HMI
Events
+
Images
ROS Navigation Stack (Localization)
Robot State
Goals
MDP/POMDP (Decision Making)
Onboard Sensors Processing (Event Detection)
Robot
ROS Navigation Stack (Localization)
Robot State
Goals
MDP/POMDP (Decision Making)
Onboard Sensors Processing (Event Detection)
Robot
Human Operator Input
Events
Events
Event Fusion and Filtering
Detections
Camera Network (Event Detection)
Server
Detections

**Figure 5.** The ROS-based implementation of the modules of our autonomous surveillance system with multiple robots.

navigate waypoints in a map. Moreover, the Decision Making module in Figure 2 is here implemented by means of MDP or POMDP planners (see the Markov Decision Making package at http://wiki.ros.org/markov_decision_making), which will be described in the "Event-Driven POMDPs for Multirobot Surveillance" section. Those planners are in charge of determining the best action for each robot and of sending the corresponding command to the navigation components.

The server macroblock is in charge of detecting events and is run on one of several physical machines wired to the network. This macroblock receives the image streams from all the cameras (including cameras on board the robots) and performs the algorithms described in the "Detecting and Disseminating Events" section to generate events. Those events are communicated to the robots and to the third macroblock, HMI, which handles all interactions with the human operators. This module is distributed into different applications. Here, we have implemented a central videowall application that allows operators to select image streams from the different cameras. Information about detected events is overlaid onto the images (as in Figure 6). We have also implemented an alternative application for mobile devices (tablets) where the operators can check events. Moreover, by interacting with a videowall displayed on their mobile devices, operators are able to send the robots to specific locations they consider relevant for surveillance. Autonomous robot surveillance is a subset of each of the following four types of activities.

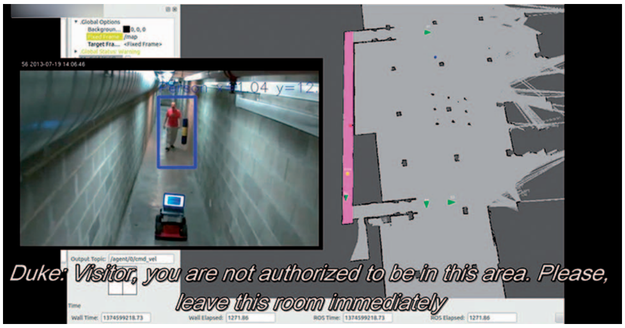

Duke: Visitor, you are not authorized to be in this area. Please, leave this room immediately

**Figure 6.** An interactive display showing the restricted zone at a location in a shopping center and a trespassing response, with the robot speaking to the intruder.

## Patrol of the Environment

The robots should maintain close surveillance of all reachable areas in the environment, paying particular attention to those most sensitive (e.g., with valuable items or not covered by static cameras). Given the dynamic nature of the environment, robots should continue to visit all areas over the course of the entire surveillance mission, not neglecting any area for too long.

## Assistance to Visitors

As noted in the "Detecting and Disseminating Events" section, the camera network can automatically detect events related to human activity, as in the case of a visitor requesting assistance (by waving to a camera). In response to such an event, one of the robots should meet the visitor and perform a simple interaction, with the intent of aiding the visitor by engaging in a simple dialog and then guiding the person to an indicated destination.

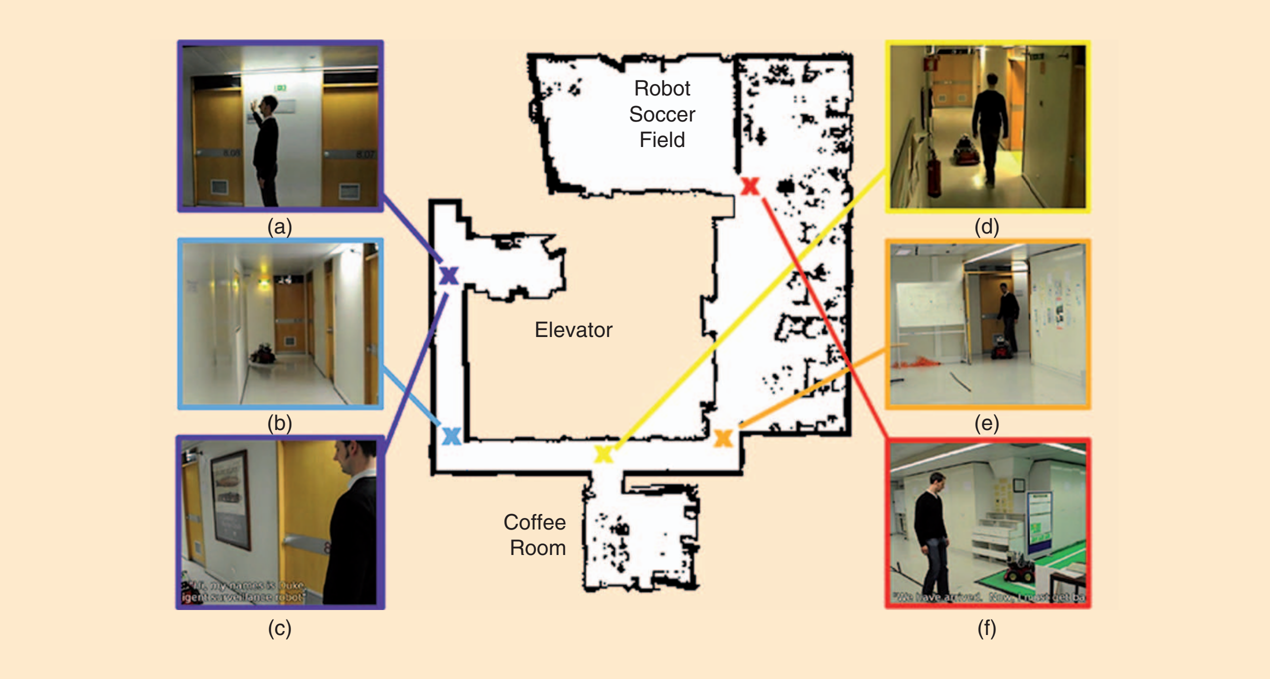


**Figure 7.** A robot's assistance to a visitor (with color coding the same as the topological graph described in Figure 4): (a) When a visitor seeks assistance (waving to a camera), (b) the robot stops patrolling and goes to the event position and (c) prompts the visitor to interact. Once the visitor tells the robot his destination, (d) and (e) the robot leads him there, (f) notifying him when the goal is reached.

### Security of Restricted Areas

Another event related to human activity is triggered whenever a trespassing person is detected in a restricted area. In this situation, one of the robots should navigate to the corresponding position of the detection and warn the trespasser, potentially alerting human security to help resolve the situation.

### Emergency Response

We also consider emergency situations that require an immediate response by the robots. For example, if a fire alarm sounds in the operating environment, robots can use additional sensors to verify whether or not the alarm is false and, if not, can even help put out the fire if capable.

## *MDPs for Single-Robot Surveillance*

In the first set of case studies, we apply an MDP technique to control a single robot following the behaviors described previously. The MDP formulation is described in the "Abstracting the System and Its Environment" section, with the robot selecting new actions whenever an event occurs or its position changes. The state space is factored into multiple variables, one for each possible event occurrence in the system (e.g., assistance requests, trespassing situations, or emergencies) and one for the position of the robot. The robot's policy is computed using an MDP model whose transition probabilities were inferred from a combination of experimental data and statistical inference and whose rewards were hand-tuned to balance the objectives. Analytical experiments have shown that the MDP approach remains tractable over long time horizons, though the performance is crucially dependent on the accuracy of bounded predictions of event likelihoods. Further details of our surveillance MDP model specifications can be found in the supplementary material accompanying this article in IEEE *Xplore*.

### Deployment in a Test Bed

First, we performed experiments in the scenario of Figure 1, which is a surveillance test bed on the floor of our research institute [2] that includes 12 static cameras, three servers, and one Pioneer 3-AT robot. The Pioneer 3-AT was a four-wheel-drive robot equipped with a SICK laser, a webcam, and speakers and programmed to navigate around the scenario, to survey remote events, and to speak warning messages. The map of the scenario, together with the corresponding topological map, is shown in Figure 4. Here, a visitor can ask for assistance by waving to the camera in the elevator hallway (as if he had just entered the floor).

Figure 7 shows a trajectory of waypoints visited by the robot during the execution of its computed policy, starting with the response to a waving event. In the absence of events, the robot behaved as expected, going around the floor and visiting all the relevant rooms. However, when the robot decided to assist a visitor who was waving, it navigated to the elevator hallway, where the waving was directly detected, without entering intermediate rooms.

We also simulated the MDP model to analyze the balance of the policy responding to surveillance events while patrolling. We ran the MDP for 100 steps, triggering fire events uniformly at random at the coffee room, and repeated 500 runs for each value of triggered fires. Figure 8 shows the percentage of extinguished fires and the number of the robot's patrol rounds. The robot performed its patrol rounds and

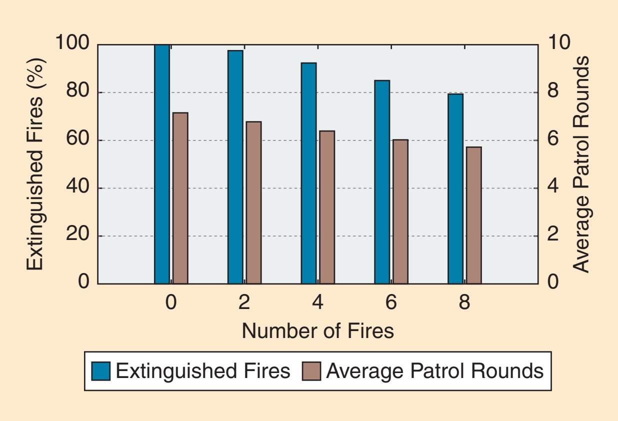


**Figure 8.** The test-bed simulations for single-robot surveillance with increasing random fire events at the coffee room. Average values for the percentage of extinguished fires and the number of patrol rounds of the robot are shown.

only stopped them to attend and extinguish fires. As expected, as there were more fires, the robot was able to perform fewer rounds. Besides, some fires had been triggered close to the end of the experiment, leaving the robot with no time to reach the coffee room. Therefore, as the number of fires increased, the extinguishing rate gradually degraded.

## Deployment in a Shopping Center

We performed a similar experiment in a more realistic environment located in a shopping mall. As a first step toward integration, we deployed our system in the technical corridors beneath the mall that are closed to the public. The map of the scenario and its topological abstraction are shown in Figure 9. Here, in addition to waving events, trespassing events were introduced. (A video summarizing the tests performed can be viewed at https://youtu.be/Ivx908SSzlk.)

In this scenario, three functionalities of the system were tested to assess its capabilities to respond to different situations using a single balanced MDP policy. In the absence of events, the robot began moving around the environment, selecting the next area to visit among those defined in Figure 9(a) and ensuring that key areas were visited frequently. During the robot's patrol, we triggered random trespassing events by entering the restricted technical corridor (see Figure 6). Each time, the robot stopped its patrol, its policy dictating that it move toward the intruder's detected position to intervene. Upon arrival, the robot requested the intruder to "leave the area immediately." After he was gone, the robot resumed its patrol. We also triggered waving events to test the robot's ability to perform visitor assistance. These tests consisted of a person entering a camera's field of view and hand-waving to request help. In response to the waving detection, the robot stopped patrolling and went to the position of the event to interact with the visitor, prompting the person to select among several possible areas in the environment. Once the visitor selected a desired destination, the robot led the way.

We carried out a third deployment of our multiagent surveillance system in the commercial, publicly accessible areas of the same shopping mall (see Figure 10). The functionalities and behaviors obtained were qualitatively identical, but the autonomous navigation of the robot was made considerably more difficult due to the characteristics of the environment and the robot's hardware limitations (e.g., the glass panes of storefronts sometimes eluded its laser range finder).

## Event-Driven POMDPs for Multirobot Surveillance

In the next experiments, we adopted an alternative decision-making approach suitable for multirobot settings with partial observability of event occurrences. In contrast to the MDP model, a POMDP explicitly considered that the event detector (and hence robots' observations) was susceptible to errors. Such errors may come in the form of false positive detections (e.g., incorrectly detecting a person in an empty room) or false negative detections (e.g., failing to detect a person).

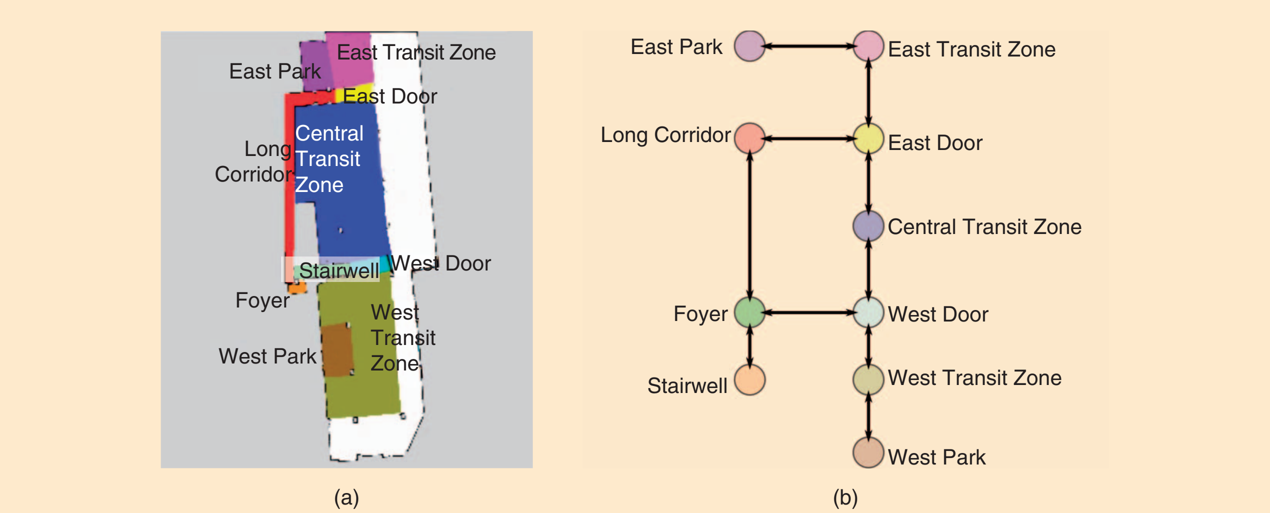


**Figure 9.** The topological map used at the shopping mall. (a) The defined selected areas designated for the robot to visit. (b) The topological connections.

Explicitly modeling observation errors, in combination with the decisions of multiple robots, comes at a computational overhead. A conventional multirobot POMDP is notoriously harder to solve than a regular MDP. Here, we circumvented the added complexity by considering the hierarchical decision-making structure shown in Figure 11. The lowest level of decision making in our system handles the navigation of each robot to its desired poses (i.e., motion planning), and this is done internally by the ROS Navigation Stack. Then, a set of tasks defines the behaviors that each robot is capable of performing individually. Each task is not necessarily bound to a particular decision-making formalism; in our case, we implemented tasks either as manually designed finite-state machines or single-robot (event-driven) POMDPs.

The cooperative decision-making problem in this scenario lies at the top of this hierarchical organization, and concerns the allocation of tasks between the robots as a response to the discrete detections of the sensor network. We cast the problem of multirobot coordination in our surveillance framework as an event-driven (asynchronous) multirobot POMDP (MPOMDP). MPOMDPs [23] are a straightforward extension of POMDPs to multirobot systems with free communication (which is the case in our surveillance system since all the robots share their information freely). As in an MDP, the POMDP model defines a set of states and actions; but it also defines a set of observations, which represent the possible incomplete or uncertain information that the robots have about their environment.

The actions in this multirobot model correspond to the abstract tasks (or behaviors in the "Common Implementation of Components" section) that each robot must perform individually: patrol of the environment, assistance to visitors (the closest robot to the visitor should respond to the event), surveillance incident response (warning trespassers in restricted areas), and emergency response. This is the highest priority task, and it should prompt robots to move to the position of the detected emergency. As with the single-robot MDP, the state space is factored into multiple variables, this time with separate variables for the local state of each robot, whether or not it is powered on, and whether or not it is busy performing a particular task (other than patrolling). As before, the rewards for each state corresponded to the relative priorities of each of the three

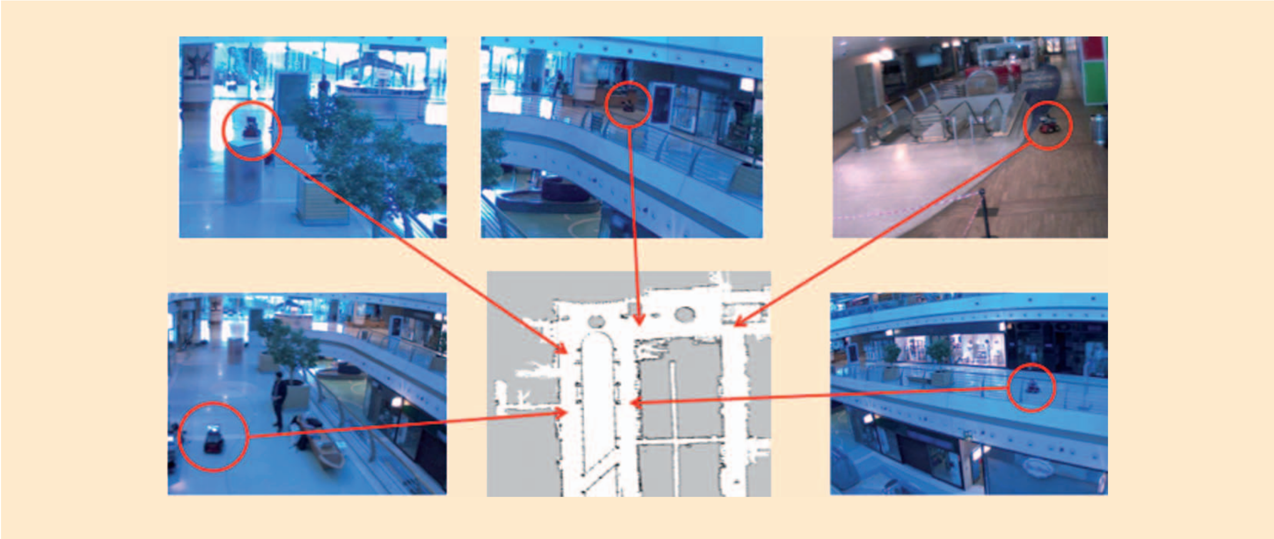


**Figure 10.** A robot patrolling public areas of the shopping mall.

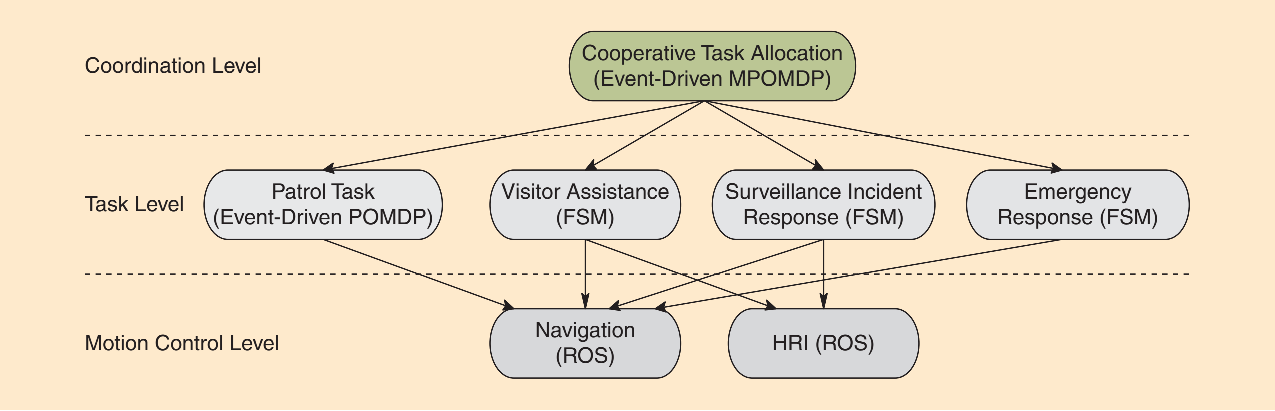


**Figure 11.** The various levels of decision making involved in our multirobot case study for autonomous surveillance. FSM: finite-state machine; HRI: human–robot interface.

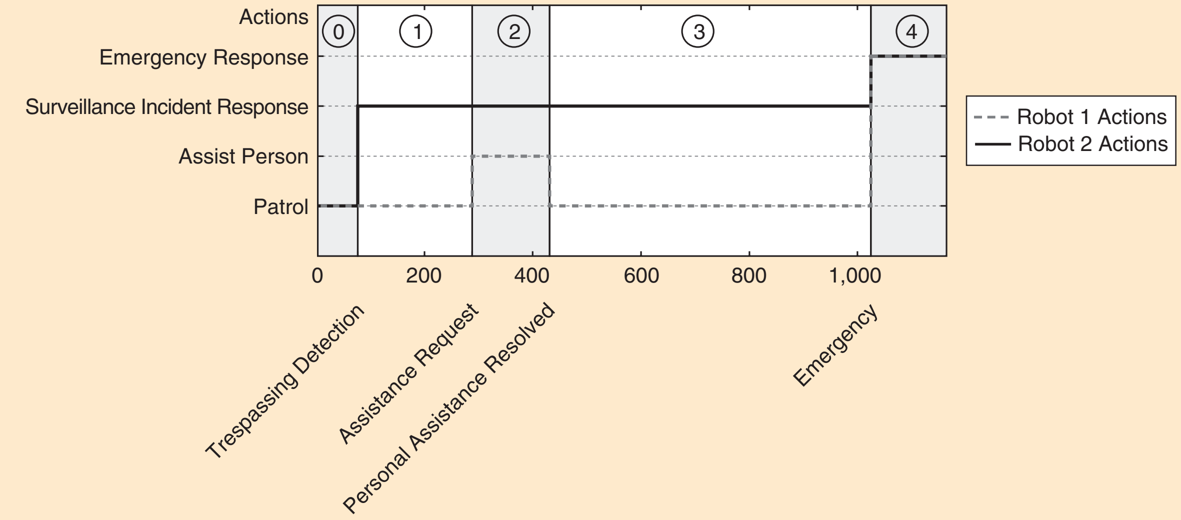


**Figure 12.** A time line of actions and events in a trial run of the multirobot case study for autonomous surveillance.

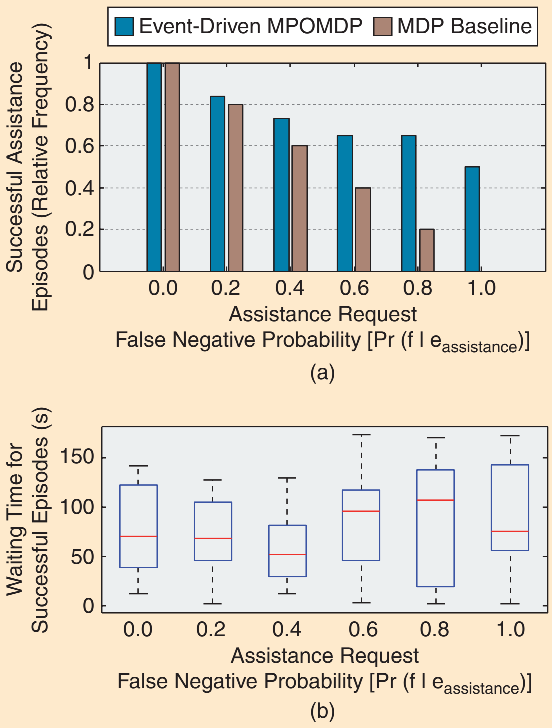


**Figure 13.** The test-bed simulations for multirobot surveillance, increasing the probability of false negative detections of assistance requests (4 h for each simulation): (a) average values of the rate of successful assistance episodes; (b) a boxplot of the visitor waiting times.

respective active events. Finally, the observations of our MPOMDP included the detection of events themselves. There was also a set of robot-specific observations (also mapped from events) that were communicated between robots so that each one could inform another of its own local state (see the supplementary material accompanying this article in IEEE *Xplore* for more details on the models).

In Figure 12, we show the timeline of a trial execution of our event-driven MPOMDP policy. That policy was computed for the same test-bed scenario described in Figure 4 but using two Pioneer 3-AT robots. In the trial, the detection of a trespasser in a restricted area prompted one robot to inspect that position by taking the surveillance incident response action at step 1. Meanwhile, the other robot continued to patrol the environment. In step 2, an assistance request was detected. Since one of the robots was already busy taking care of the trespasser, the remaining robot (robot 1) decided to assist the visitor. Afterward, the robot went back to patrolling the environment until, at step 4, a fire detection was simulated, which caused both robots to abandon their active tasks and address the emergency immediately. The total runtime of this trial (19 min, 18 s) was limited only by the battery lifetime of each robot.

Figure 13 depicts simulation results to assess our event-driven MPOMDP policy for the assistance of visitors. We performed experiments of fixed time length (4 h each) while increasing the probability of false negative detections, i.e., failing to detect visitor assistance requests. Then we measured the rate of successful visitor assistance episodes and the waiting times for those, for both the event-driven POMDP as well as for a baseline MDP (that assumes full observability). The results showed that, as the probability of false negatives increased (and therefore the reliability of the camera network decreased), the POMDP policy was able to successfully respond to more assistance requests than the MDP baseline, since the former explicitly considered observations as stochastic, and reasoned over the possibility that an undetected person was waiting for assistance. Even with complete unobservability (i.e., without ever being able to observe a request for assistance through the camera network), the POMDP policy still drove the robot to periodically check for any possible visitors. In Figure 13(b), the waiting times for assisted visitors are also shown to be relatively independent of the reliability of the sensors, as there was no statistically significant difference between the respective distributions. This means that the POMDP policy induces an efficient patrol strategy that minimizes the risk that a visitor is left waiting for too long.

### Limitations and Extensibility

The prototype deployments documented in the preceding sections provide proof of concept on which future studies can build and extend beyond the system's present limitations. These limitations include, for instance, the number of robots, the richness of scenarios, and the scope of the deployment. These are not indicative of shortcomings of the surveillance framework itself, but are rather due to the limited resources over the relatively short term that this project was carried out. Given substantial supplemental support as well as the necessary permissions, a natural next step would be to operate the surveillance robots in public areas of the shopping center, leading to a more comprehensive evaluation of the performance of the system as a whole.

One might also consider limitations imposed by the robots' decision-theoretic planning methods. For instance, POMDPs have the reputation of being hard to scale. Fortunately, we can mitigate the computational increase commonly associated with adding more robots or surveilling larger areas by employing recent research advances, such as factored models [11], [21], decoupling [28], and hierarchical planning [1], [17]. More advanced methods following these paradigms are well accommodated by the surveillance framework, which already has the capacity to decentralize the robots' planning and awareness and to represent surveillance tasks with varying degrees of abstraction. In particular, note that we exploited in our case studies both factored and hierarchical models (see the supplementary material accompanying this article in IEEE *Xplore* for more details).

Another challenge that could be perceived as a limitation of the current methods used to make robot surveillance decisions is the specification of effective MDP parameters (i.e., state feature, transition probabilities, and rewards). Such models are general enough to induce the complex behavioral policies that we have demonstrated and a wide variety of other robot behaviors. However, prescribing accurate probabilities is easier said than done in a real surveillance environment outside of the laboratory, where we have the limited ability to collect data with the real robots. This has since led us to consider more sophisticated modeling techniques that employ statistical inference on easy-to-collect parameters to help derive reasonable settings for hard-to-collect parameters [29]. Similarly, we have found it nontrivial to select rewards that adequately balance competing surveillance objectives. Though preliminary advances have been made, these issues warrant further research.

## Conclusions

The framework we have developed constitutes an important step toward fully autonomous surveillance. We introduce into the conventional surveillance system mobile robots that have the potential to alleviate the tasks of human operators. Our robots embody intelligent surveillance nodes capable of pursuing a variety of surveillance activities and of deciding among activities in real time based on the occurrence and urgency of events in a dynamic and uncertain environment. Underlying the robots' autonomy is a framework architecture that automatically detects anomalies, aggregates and filters detections to interpret them as events, transmits those events to the robot, and responds by intelligent reasoning, navigation, and physical interaction.

This is all made possible by leveraging several complementary research technologies, such as computer vision, robot automation, and intelligent decision making, and integrating them into a cohesive, modular design. Our case studies demonstrate a progression toward increasingly complex scenarios in increasingly realistic surveillance environments, whereby we have been able to take our system out of the laboratory and into a shopping center.

However, the primary benefit of our framework is that it serves as a research platform with which to apply decision-making formalisms and techniques to real robot problems. Autonomous surveillance is a rich domain wherein resource constraints, uncertainties, and competing objectives provide significant challenges that can be addressed through decision-theoretic planning. This has driven us to develop solutions using MDPs and POMDPs as described in our case studies, pushing the state of the art and developing novel advances for planning in real-world settings [17], [18], [29].

## Acknowledgments

This work was partially supported by the Portuguese Fundação para a Ciência e a Tecnologia (FCT), through strategic funding for Institute for Systems and Robotics/Laboratory for Robotics and Engineering Systems (ISR/LARSyS) under grant PEst-OE/EEI/LA0021/2013 and through the Carnegie Mellon Portugal Program under grant CMU-PT/SIA/0023/2009. This study also received national funds through the FCT, with reference UID/CEC/50021/2013, and through grant FCT UID/EEA/50009/2013 of ISR/LARSyS.

## References

[1] C. Amato, G. D. Konidaris, and L. P. Kaelbling, "Planning with macro-actions in decentralized POMDPs," in *Proc. Int. Conf. Autonomous Agents and Multiagent Systems*, 2014, pp. 1273–1280.

[2] M. Barbosa, A. Bernardino, D. Figueira, J. Gaspar, N. Goncalves, P. Lima, P. Moreno, A. Pahliani, J. Santos-Victor, M. Spaan, and J. Sequeira, 2009. "Isrobotnet: A testbed for sensor and robot network systems," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2009, pp. 2827–2833.

[3] R. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1957.

[4] P. Chen, P. Ahammad, C. Boyer, S. I. Huang, L. Lin, E. Lobaton, M. Meingast, S. Oh, S. Wang, P. Yan, A. Yang, C. Yeo, L. C. Chang, J. Tygar, and S. Sastry, "Citric: A low-bandwidth wireless camera network platform," in *Proc. 2nd ACM/IEEE Int. Conf. Distributed Smart Cameras (ICDSC)*, 2008, pp. 1–10.

[5] R. Cucchiara, C. Grana, A. Prati, G. Tardini, and R. Vezzani, "Using computer vision techniques for dangerous situation detection in domotic applications," in *Proc. IEE Intelligent Distributed Surveilliance Systems*, 2004, pp. 1–5.

[6] A. del Corte, O. Gutierrez, and J. Gomez, "New location techniques based on ray-tracing for increasing airport safety in apron and stand areas," *Frontiers in Computer Education*, S. Sambath and E. Zhu, Eds. Berlin: Springer-Verlag, 2012, pp. 515–522.

[7] F. Delle Fave, S. Canu, L. Iocchi, D. Nardi, and V. Ziparo, "Multi-objective multi-robot surveillance," in *Proc. 4th International Conference on Autonomous Robots and Agents (ICARA)*, 2009, pp. 68–73.

[8] A. Fernández-Caballero, J. C. Castillo, M. T. López, J. Serrano-Cuerda, and M. V. Sokolova, "INT3-Horus framework for multispectrum activity interpretation in intelligent environments," *Expert Syst. with Applicat.*, vol. 40, no. 17, pp. 6715–6727, 2013.

[9] C. Galindo, J. A. Fernández-Madrigal, J. González, and A. Saffiotti, "Robot task planning using semantic maps," *Robotics and Auton. Syst.*, vol. 56, no. 11, pp. 955–966, 2008.

[10] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *Int. J. Robotics Res.*, vol. 23, no. 9, pp. 939–954, 2004.

[11] C. Guestrin, D. Koller, and R. Parr, "Multiagent planning with factored MDPs," in *Proc. Advances in Neural Information Processing Systems (NIPS),* Vancouver, Canada, 2001, pp. 1523–1530.

[12] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," in *Proc. 10th IEEE Int. Conf.Computer Vision (ICCV)*, 2005, vol. 1, pp. 166–173.

[13] Y. Ke, R. Sukthankar, and M. Hebert, "Spatio-temporal shape and flow correlation for action recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[14] S. Kullback and R. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[15] T. Lemaire, R. Alami, and S. Lacroix, "A distributed tasks allocation scheme in multi-UAV context," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA),* 2004, vol. 4, pp. 3622–3627.

[16] R. T. Maheswaran, M. Tambe, E. Bowring, J. P. Pearce, and P. Varakantham, "Taking DCOP to the real world: Efficient complete solutions for distributed multi-event scheduling," in *Proc. 3rd Int. Joint Conf. Autonomous Agents and Multiagent Systems*, vol. 1, 2004, pp. 310–317.

[17] J. Messias, M. Spaan, and P. Lima, "GSMDPs for multi-robot sequential decision-making," in *Proc. AAAI Conf. Artificial Intelligence*, 2013, pp. 1408–1414.

[18] J. Messias, "Decision-making under uncertainty for real robot teams," Ph.D. dissertation, Instituto Superior Técnico, Univ. Lisbon, Portugal, 2014.

[19] P. Moreno, A. Bernardino, and J. Santos-Victor, "Waving detection using the local temporal consistency of flow-based features for real-time applications," in *Image Analysis and Recognition*, M. Kamel and A. Campilho, Eds. Berlin: Springer-Verlag, 2009, pp. 886–895.

[20] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *Int. J. Comput. Vision*, vol. 79, no. 3, pp. 299–318, 2008.

[21] F. A. Oliehoek, S. Whiteson, and M. T. J. Spaan, "Approximate solutions for factored Dec-POMDPs with many agents," in *Proc. 12th Int. Conf. Autonomous Agents and Multiagent Systems (AAMAS)*, 2013, pp. 563–570.

[22] V. Onut, D. Aldridge, M. Mindel, and S. Perelgut, "Smart surveillance system applications," in *Proc. Conf. Center for Advanced Studies on Collaborative Research*, 2010, pp. 430–432.

[23] D. V. Pynadath and M. Tambe, "The communicative multiagent team decision problem: Analyzing teamwork theories and models," *J. Artificial Intell. Res.,* vol. 16, no. 1, 2002, pp. 389–423.

[24] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *Proc. Open-Source Software Workshop Int. Conf. Robotics and Automation*, Kobe, Japan, 2009, pp. 1–6.

[25] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th Int. Conf. Pattern Recognition (ICPR),* vol. 3, pp. 32–36.

[26] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable Markov processes over a finite horizon," *Operations Res.*, vol. 21, no. 5, pp. 1071–1088, 1973.

[27] O. Svenonius, "The Stockholm Security Project: Plural policing, security and surveillance," *Inform. Polity*, vol. 17, no. 1, pp. 35–43, 2012.

[28] S. J. Witwicki and E. H. Durfee, "Towards a unifying characterization for quantifying weak coupling in Dec-POMDPs," in *Proc. 10th Int. Conf. Autonomous Agents and Multiagent Systems*, 2011, vol. 1, pp. 29–36.

[29] S. J. Witwicki, F. S. Melo, J. Capitán, and M. T. Spaan, "A flexible approach to modeling unpredictable events in MDPs," in *Proc. 23rd Int. Conf. Automated Planning and Scheduling (ICAPS)*, 2013, pp. 260–268.

[30] Y. Yao, C. H. Chen, A. Koschan, and M. Abidi, "Adaptive online camera coordination for multi-camera multi-target surveillance," *Comput. Vision and Image Understanding*, vol. 114, no. 4, pp. 463–474, 2010.

***Stefan Witwicki***, Robotic Systems Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland. E-mail: witwicki@gmail.com.

***José Carlos Castillo***, Department of Systems Engineering and Automation, Carlos III University, Madrid, Spain. E-mail: jocastil@ing.uc3m.es.

***João Messias***, Intelligent Systems Laboratory, University of Amsterdam, The Netherlands. E-mail: jmessias@uva.nl.

***Jesús Capitán***, Robotics, Vision, and Control Group, University of Seville, Spain. E-mail: jcapitan@us.es.

***Francisco S. Melo***, Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento em Lisboa and Instituto Superior Técnico, University of Lisbon, Portugal. E-mail: fmelo@inesc-id.pt.

***Pedro U. Lima***, Institute for Systems and Robotics, Instituto Superior Técnico, University of Lisbon, Portugal. E-mail: pal@isr.tecnico.ulisboa.pt.

***Manuela Veloso***, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, United States. E-mail: mmv@cs.cmu.edu.