A Dynamic Bayesian Network Approach to Surveillance Decision Planning

Leon Chong and Raymond Jarvis
Intelligent Robotics Research
Department of Electrical and Computer Systems Engineering
Monash University, Clayton, Australia
{chow.chong,ray.jarvis}@eng.monash.edu.au

Abstract

This paper explores the use of Dynamic Bayesian Networks as a decision process to aid a mobile robot in planning its surveillance route. The primary concern of mobile surveillance robots is to determine the optimal route that will reduce the risk of not observing a relevant surveillance event. To aid the robot's decision process, a popular approach is to model the environment with prior probabilities of observing relevant events and utility values to represent the penalty costs of relevant events that remain undetected. By exploiting the general layout of indoor office buildings, a Dynamic Bayesian Network is applied to model the decision planning task as a Bayesian estimation problem. Combined with the prior and utility value, the expected cost for each segment of the building is obtained from the Bayesian estimate to aid in planning an optimal route for the mobile robot.

1 Introduction

The role of mobile surveillance robots is to continually observe a designated environment and to record occuring surveillance events. The mobility of robots would not restrict surveillance efforts to a small area of the environment, instead sensors that are mounted on these robots would act as mobile surveillance sensors that could span the entirety of a designated environment. The algorithm developed in this paper is primarily designed for security surveillance applications, such as the detection of unauthorised persons in an indoor office building.

Surveillance effort begins by finding the occurance of a relevant event for the purpose of recording and analysis. In the case of security surveillance applications, this may involve searching the environment space for potential intruders. It can be argued that this is the most critical part of the design of mobile surveillance systems, that is to be able to increase the chance of a mobile surveillance system to discover a relevant surveillance event that is occuring. A paper by [Massios and Voorbraak, 1998] addresses this problem as the primary concern of mobile surveillance systems. By modeling the problem with decision-theory, an optimal route for a mobile surveillance robot could be planned to reduce the chance of a relevant event remaining undetected [Massios and Voorbraak, 1998]. This approach models the environment with a set of prior probabilities to indicate the known probabilites of each segment of a designated environment that a relevant event will occur. Utility values are assigned as well to indicate the importance of finding a relevant event in a particular area. Consequently, a minimum expected cost policy is used to plan the sequence of surveillance goals. A significant weakness of the aforementioned surveillance decision algorithm is that the travelling time between areas of a designated environment is neglected. Therefore, the algorithm does not optimise the navigation route based on the locality of each area, rather the algorithm minimises only the total expected cost of the environment.

A follow-up paper [Massios *et al.*, 2001] redefines the solution to take advantage of the environment's geometry. Clustering of multiple regions is performed by exploiting the geometry of indoor office blocks. This allows neighbouring regions to share common expected costs at multiple clustering levels. The algorithm proposed in this paper is similar to the clustering method, in which neighbouring regions are clustered to form common nodes. However, instead of producing clusters based on the geometry structure, the proposed algorithm works by indentifying local neighbours to produce a Directed Acyclic Graph (DAG).

The produced DAG is used to form a Bayesian Network to approximate the posterior probability of neighbouring regions. It is assumed that the prior probabilities of critical regions are known. By taking into account the dependence between neighbouring regions, conditional probability will dictate that observations that are made at a particular area will influence the probability of its neighbouring regions. With that assumption, a surveillance robot is required only to observe a reduced set of regions in a cluster in order to deduce the probability of an event occuring in its neighbours. This helps in structuring more optimal navigational routes as a robot could visit or pass through a lesser set of regions to update the event probability of its neighbours.

The rest of this paper describes the reason and solution of using Dynamic Bayesian Networks as a viable algorithm to decision planning for a security surveillance system. The modelling of an indoor office building is explained as well as how the prior probabilities and utility values are used in the decision process. Experimental results from a simulated environment are presented, followed by a discussion on future extensions to the algorithm.

2 Dynamic Bayesian Network

The general structure of indoor office buildings allows the use of a DAG to delineate the relationship between individual segments of the building. With regards to security surveillance applications, this relational structure accounts for the probability dependency between neighbouring segments of the building when relevant surveillance events occur. Based on the required criteria, the Dynamic Bayesian Network (DBN) algorithm was selected to model the conditional probabilities of neighbouring segments. This approach allows the probability of each individual segment to be updated based on spatial and temporal observations made.

2.1 Environment Segmenting

The general layout of indoor office buildings is used as a guideline to segment the environment (refer to Figure 1). Each segment consists of either a room or corridor and each segment are represented as a node in the DAG. Arcs, as part of the DAG, are formed between each segments to indicate local neighbours.

Assuming each room are approximately constant in size, it will be suffice to represent each room as a single node in the graph. Room nodes are defined as $R^{(i)}, i \in \{1, \ldots, N_r\}$, where N_r is the number of rooms in the environment. Rooms are identified based on the criteria that a room is a significant surveillance goal, that is a room has to have a prior probability and a utility value. Based on that assumption, it can be surmised that the primary goal of security surveillance systems is to be able to reach and survey the rooms of an environment in a timely manner.

Corridors of an environment are merely a connecting platform between the various rooms. A corridor can be divided into multiple segments depending on its physical size. Each corridor segment are represented by a single node in the graph and are defined as $C^{(k)}, k \in \{1, \ldots, N_c\}$, where N_c is the number of corridor segments in the environment. As its sole purpose is to connect multiple rooms, corridors are not associated with prior probabilities. However, corridors with rooms as their local neighbours would share the combined sum of utility values of their neighbouring rooms.

As critical surveillance goals consist of adjoining rooms with known prior probabilities, room nodes $R^{(i)}$ are fixed as root nodes in the DAG. Therefore, directed arcs are formed between room nodes and corridor nodes in which room nodes are directed as the parents. This configuration would allow the use of a DBN to infer the belief of corridors conditioned upon the prior and observation of their neighbouring room nodes.

2.2 Local Neighbourhood Dependency

The primary goal of Bayesian Networks is to infer the belief of a node conditioned upon the observations and/or beliefs of its neighbouring nodes. This conditional relationship will result in various dependency cases for different node observations and connectivity structures. These conditions are illustrated in [Murphy, 1998]. Consider a network consisting of three nodes, $R^{(1)} \longrightarrow C^{(1)} \longleftarrow R^{(2)}$. In all cases, the belief of $C^{(1)}$ is dependent on nodes $R^{(1)}$ and $R^{(2)}$.

If none of the nodes are observed (i.e. all nodes are hidden), $R^{(1)}$ and $R^{(2)}$ are independent of each other. This is the most trivial case which will result in the belief of $C^{(1)}$ being conditioned upon the prior probabilities of $R^{(1)}$ and $R^{(2)}$. In terms of surveillance decision planning, this will result in the probability of observing a relevant security event in corridor $C^{(1)}$ to be dependent on the prior probabilities of it's neighbouring rooms.

If either $R^{(1)}$ or $R^{(2)}$ is observed, the belief of $C^{(1)}$ will be updated, however, the belief of the hidden room node will remain unchanged. The belief of $C^{(1)}$ will now reflect the observation that was made in one of the rooms. If an intruder is observed, the belief of $C^{(1)}$ will increase proportionate to the belief of the hidden room. Conversely, the the belief of $C^{(1)}$ will decrease if an intruder is not observed.

If node $C^{(1)}$ is observed, the belief of nodes $R^{(1)}$ and $R^{(2)}$ will be updated. The belief of nodes $R^{(1)}$ and $R^{(2)}$ will now reflect the updated belief of observing an intruder in the rooms. This is akin to predicting the room an intruder will enter once he is observed in the neighbouring corridor. The updated belief is not meant to be treated as an accurate prediction of the intruder's planned path, however, it is used as a relative measure to determine the likelihood of the intruder to enter a particular room rather than advancing to other rooms or corridors.

By assuming two observations can be made almost simultaneously, if node $C^{(1)}$ is observed and any one of the the rooms nodes $R^{(1)}$ or $R^{(2)}$ are observed as well, then nodes $R^{(1)}$ and $R^{(2)}$ will be dependent on each other. This will update the belief of the hidden room node based on the resulting events of the observed nodes. For example, if an intruder is observed at nodes $C^{(1)}$ and $R^{(1)}$, then the belief of room $R^{(2)}$ of observing an intruder will decrease. This resultant scenerio is quite intuitive, if the intruder is found in room $R^{(1)}$, then the likelihood of observing an intruder in room $R^{(2)}$ will be reduced. The converse scenario is true as well, in which an intruder that is not observed in room $R^{(1)}$ will result in an incresed likelihood of observing an intruder in room $R^{(2)}$.

It is important to note that the above scenarios do not accurately depict the general characteristics of Bayesian Networks. These scenarios are only achieved by careful modelling of the conditional probability table (CPT) for the security surveillance application. Also, the beliefs obtained from the Bayesian Network are not exact inference of observing relevant security events. Instead, the belief values are used as relative guildelines to determine the likelihood of observing relevant security events in various parts of the building.

2.3 Temporal Dependency

Apart from spatial dependency, the system needs to exhibit temporal dependency as well. The use of a single mobile robot for surveillance applications will dictate that only one segment of the environment can be surveyed at a single time instance. Therefore, an observation made at time t will be invalid at time $t + \delta$, assuming the robot is not stationary. The time constant, δ, indicates the reasonable amount of time that an observation should remain valid after leaving the area. This is to ensure that an observation will not be invalid immediately after leaving the observed area. Instead, it is a gradual process which will update the area's belief for the duration of time, δ.

DBNs can be used to model the required temporal dependency system. The prior and interslice connectivity are modelled based on the Bayesian Network structure in Section 2.1. To create a dynamic system, intraslice arcs are added. Since each segment of the environment will be time dependent, arcs are formed between $R_t^{(i)} \longrightarrow R_{t+1}^{(i)}$ and $C_t^{(k)} \longrightarrow C_{t+1}^{(k)}$, where $t \in \{0, \ldots, T\}$ is the current time instance. Hence, the belief of a node at time t will be dependent on the beliefs and observations of that node at previous time instaces $0, \ldots, t-1$:

$$Bel(R_t^{(i)}) = P(R_t^{(i)} | R_{t-1}^{(i)} \ldots R_0^{(i)})$$

$$Bel(C_t^{(k)}) = P(C_t^{(k)} | R_t^{(k_1)} \ldots R_t^{(k_m)} C_{t-1}^{(k)} \ldots C_0^{(k)})$$

Room Observations	$P(C_0^{(k)} = T \| R_0^{(k_1)} \ldots R_0^{(k_m)})$
All True	0.99
All False	0.1
Half True, Half False	0.7

Table 1: A corridor's prior CPT for the trivial room observation cases, where $R_0^{(k_1)} \ldots R_0^{(k_m)}$ are the m room neighbours of $C_0^{(k)}$.

where $R_0^{(k)}$ is the prior probability for node $R^{(k)}$, $C_0^{(k)}$ is the prior probability for node $C^{(k)}$ and $R_t^{(k_1)} \ldots R_t^{(k_m)}$ are the m room neighbours of node $C_t^{(k)}$.

3 Probability Setting

3.1 Prior CPT

The state of each node is observed as either "true" or "false" ("true" if an intruder is found and "false" otherwise). Hence, each node is discrete and consists of two values. As rooms are defined as root nodes, their prior states depend only on their prior probability:

$$Bel(R_0^{(i)} = true) = P(R_0^{(i)})$$

Conversely,

$$Bel(R_0^{(i)} = false) = 1 - P(R_0^{(i)})$$

Prior state calculation for corridor nodes are more complicated as the CPT needs to be defined. As there are no known solution to build the CPT, a heuristic interpretation of the problem is used to build it. The size of the CPT for a corridor node depends on the number of connected room neighbours. In general, the number of rows of a corridor's CPT is 2^p where p is the number of room neighbours. The most trivial entries for the CPT are defined in Table 1. To create a redundant system to ensure that all relevant events will be captured, the CPT value is made more true than false to ensure that when a node is observed to be false, there will still be a higher chance that it is true.

The rest of the CPT entries are defined using the following formula:

$$CPT_{xTyF}^{C_0^{(k)}=T} = 0.7 + \frac{0.3}{2^y}, \text{if } x > y$$

$$CPT_{xTyF}^{C_0^{(k)}=T} = 0.7 - \frac{0.2}{2^x}, \text{if } x < y$$

where $CPT_{xTyF}^{C_0^{(k)}=T}$ is the CPT row table entry for corridor $P(C_0^{(k)} = T | R_0^{(k_1)} \ldots R_0^{(k_m)})$ when x of its room neighbours are observed to be "true" and y of its room neighbours are obseved to be "false". Again, more emphasis is given to the true case than false. The values that were chosen are based on experimental results.

3.2 Intraspace Node

An important approximation of the Bayesian Network that has not been mentioned so far is the interconnectivity between the building's corridors. With directed arcs, it is difficult to model the corridor's interconnectivity as there is no clear approach to determine the direction of the arc between corridors. One way to solve this problem is to divide each corridor into individual networks and to introduce intraspace nodes to approximate the interconnectivity between each corridor.

This approach would result in the creation of multiple Bayesian Networks. At first glance, it may seem that doing so would increase the computational complexity of the system. However, to the contrary, the complexity of the system will be significantly reduced as compared to a single Bayesian Network system. This is due to the complexity of the forwards-backwards algorithm that is used as an exact inference algorithm for Bayesian Networks, which will result in a HMM with $S = M^{N_h}$ states, where M is the number of values per node and $N_h \simeq N_r + N_c$ is the number of hidden nodes [Murphy, 2002]. By dividing the system into multiple networks, the number of HMM states for infering the Bayesian Network would now be $S = \sum_{i=1}^{N_c} M^{N_{r_i}}$ where N_{r_i} is the number of room neighbours of corridor $C^{(i)}$.

The role of the intraspace node is to approximate the current belief of the neighbouring corridors. Therefore, an intraspace node is added in place of each neighbouring corridor of the network. To approximate the dependency of each corridor node to their corridor neighbours, intraspace nodes are directed as the parents for each node. As a further reduction to the computation complexity, the values of intraspace nodes are discrete to represent their corresponding corridor's beliefs in discrete intervals.

With the introduction of intraspace nodes, the CPT for each corridor needs to be to redefined. To illustrate the construction of a CPT with intraspace nodes, assume the connectivity of the following four node system, $R^{(2)} \rightarrow C^{(2)} - C^{(1)} \leftarrow R^{(1)}$. To build the Bayesian Network for nodes $C^{(1)}, R^{(1)}$, the following network is constructed to simulated the undirected arc between nodes $C^{(2)}$ and $C^{(1)}$, $I_{C^{(2)}} \rightarrow C^{(1)} \leftarrow R^{(1)}$, where $I_{C^{(2)}}$ is the intraspace node to represent the belief of corridor $C^{(2)}$. Similarly, the structure of the Bayesian Network for nodes $C^{(2)}, R^{(2)}$, will be $R^{(2)} \rightarrow C^{(2)} \leftarrow I_{C^{(1)}}$. Only taking into account the network of $C^{(1)}$, the interface node $I_{C^{(2)}}$ needs to approximate the belief of $C^{(2)}$ when $I_{C^{(2)}}$ is observed. Therefore, the belief of $C^{(1)}$ is to be approximated as:

$$\begin{aligned} Bel(C^{(1)} = T) &= \sum_j \sum_k P(C^{(1)} = T | R_j^{(1)}, C_k^{(2)}) \times \\ & \quad Bel(R_j^{(1)}) Bel(C_k^{(2)}) \\ &\simeq \sum_j P(C^{(1)} = T | R_j^{(1)}, I_{C^{(2)}} = x) \times \\ & \quad Bel(R_j^{(1)}) \end{aligned}$$

where x is the current belief of node $C^{(2)}$. Hence, the modified CPT, $P(C^{(1)} = T | R_j^{(1)} = y, I_{C^{(2)}} = x)$, where y is the state of $R_j^{(1)}$, is:

$$\begin{aligned} & P(C^{(1)} = T | R_j^{(1)} = y, I_{C^{(2)}} = x) = \\ & \quad P(C^{(1)} = T | R^{(1)} = y, C^{(2)} = T) x + \\ & \quad P(C^{(1)} = T | R^{(1)} = y, C^{(2)} = F)(1 - x) \end{aligned}$$

Therefore, the generalised prior CPT can be obtained with the following equation:

$$\begin{aligned} P(C_0^{(k)} = T \quad | & \quad R_0^{(i_1)} = y_{i_1} \ldots R_0^{(i_m)} = y_{i_m}, \\ & \quad I_{C^{(k_1)}} = x_{k_1} \ldots I_{C^{(k_n)}} = x_{k_n}) = \\ & \sum_{i_1=0}^{1} \cdots \sum_{i_n=0}^{1} CPT_{xTyF}^{C_0^{(k)}=T} . V_{k_1}^{i_1} \ldots V_{k_n}^{i_n} \end{aligned}$$

where $V_{k_i}^0 = 1 - x_{k_i}$ and $V_{k_i}^1 = x_{k_i}$, $C^{(k_1)} \ldots C^{(k_n)}$ are the n corridor neighbours of $C^{(k)}$ with $x_{k_1} \ldots x_{k_n}$ their corresponding discretised beliefs. $CPT_{xTyF}^{C_0^{(k)}=T}$ can be obtained from Section 3.1 with the corridor neighbours included as part of x and y.

3.3 DBN Probability Setting

For temporal dependency, DBNs are used to model each network in the system. DBNs will require additional modification to the CPT as each node are now dependent on the beliefs and observations of previous nodes. This additional dependency introduces new parent arcs between the current interslice nodes and the previous interslice nodes. Temporal arcs are formed only between nodes of the same type, i.e. they are formed between the following nodes $R_t^{(i)} \longrightarrow R_{t+1}^{(i)}$, $C_t^{(k)} \longrightarrow C_{t+1}^{(k)}$ and $I_{C_t^{(k)}} \longrightarrow I_{C_{t+1}^{(k)}}$.

The primary purpose of modeling the system with temporal dependency is to ensure that observations that are made at time t are not valid after time $t + \delta$. To implement such a system, the following equations are used to build the CPT for the intraslice nodes:

$$P(Q_1 = x | Q_0 = \neg x, parents) = \frac{P(Q_0 = x | parents)}{\gamma}$$

$$P(Q_1 = \neg x | Q_0 = x, parents) = \frac{P(Q_0 = \neg x | parents)}{\gamma}$$

where Q_1 is the node immediately after the prior timeslice, Q_0, x are the possible values of node Q and γ is a dampening factor.

The defined CPT will ensure that after time $t + \delta$ since an observation at time t was made, the belief of node $Q_{t+delta}$ will reduce to the prior belief of Q_{t-1}. The dampening factor, γ, will ensure that the belief will gradually increase or decrease from time t to time $t + \delta$. Hence, $\delta \propto \gamma$.

4 Decision Planning

Based on the maximum expected cost policy, surveillance routes are planned by searching the environment space for rooms and corridors with the highest expected cost. The expected cost for each part of the building is calculated using the following formula:

$$EC_t^{Q^{(i)}} = Bel(Q_t^{(i)}).U(Q^{(i)})$$

where $Bel(Q_t^{(i)})$ is the current belief of node $Q^{(i)}$ as obtained from the DBN at time t and $U(Q^{(i)})$ is the utility of node $Q^{(i)}$.

The maximum expected cost policy will always select surveillance goals with the highest probability of a relevant security event occuring and the highest utility value. Utility values indicate the importance of finding a relevant event. Therefore high utility values will signify areas that are of high importance.

Expected costs are updated as beliefs are updated from observations at various parts of the building. Since expected costs depend on the approximate posterior probabilites, areas with high prior probabilities and high utility values would be less important if no intruder is observed in these areas. This would allow other surveillance goals that have not been recently observed to be selected as part of the robot's next route. Due to temporal dependencies, observed nodes will regain thier prior importance after time $t + \delta$. Therefore, nodes that have not been observed for some time will be revisited.

The utility values of corridor nodes are set as the sum of the utility values of their neighbouring rooms. The consequence of this is that corridor nodes will be of more importance than their neighbouring room nodes, if the corridor nodes remain unobserved. This will cause the mobile robot to visit the corridor of rooms more frequently than visiting the rooms themselves. The intuition behind this is that in a large office building, there is insufficient time for a robot to visit each individual room. Instead, it will be more efficient to visit the neighbouring corridors of each room initially and then to visit individual rooms that are more important than the cluster of neighbouring rooms at a later stage. This scenario will eventually arise because as corridor nodes are observed, the beliefs of their neighbouring rooms will decline but not as much as the observed corridor. The eventuality of this is that the unobserved room nodes will obtain a higher expected cost value than their neighbouring corridor nodes.

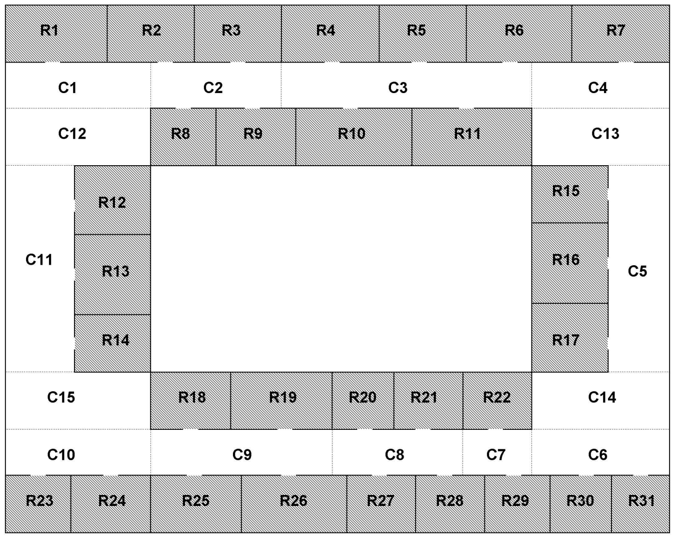

Figure 1: Layout of the simulated environment.

An overview of the decision planning algorithm is summarised below:

1. Obtain the marginal probability for each room and corridor node.
2. Calculate the expected cost for each node.
3. Find the best route to the room or corridor with the highest expected cost.
4. Update the observation of the target node when it is in the target area.
5. Update the intraspace nodes of the target node's neighbouring networks.
6. Hide the target node and repeat Step 1.

5 Experimental Results

To analyse the decision planning algorithm, a simulated environment was created. The layout of the hypothetical environment is illustrated in Figure 1. Two algorithms, the DBN algorithm and the decision-theoretic algorithm [Massios and Voorbraak, 1998], were implemented and compared.

The simulated environment was segmented manually to a total of 31 rooms and 15 corridors, all of which are associated with random prior probabilities and utility values. With the DBN decision planning algorithm, the number of DBNs that were created was 11. One DBN was created for each corriddor with neighbouring rooms. The total number of nodes that were created, including intraspace nodes, was 61. The dampening factor, γ, was set to 1000 for this simulation.

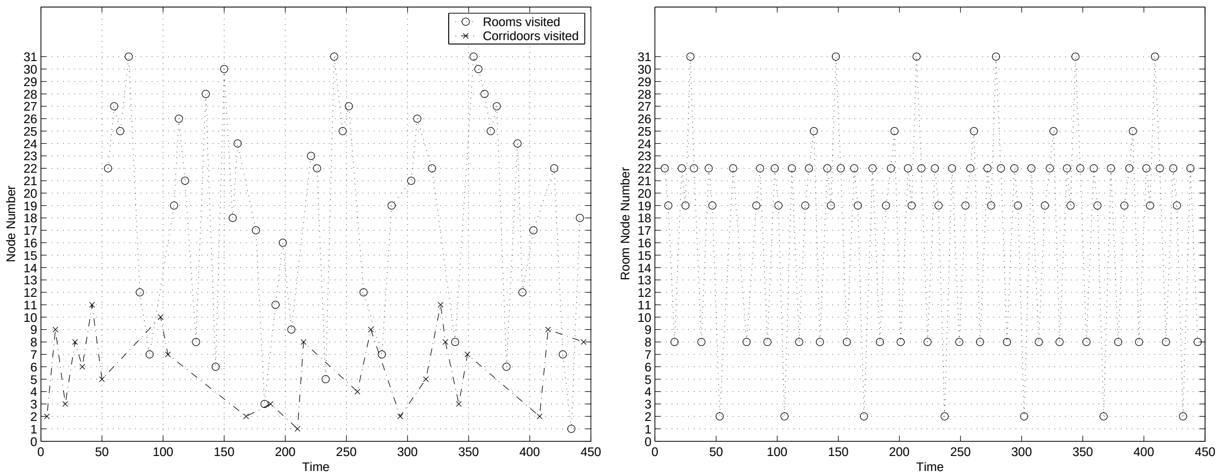

Figure 2: The figure to the left illustrates the surveillance routes that were chosen by the DBN decision planning algorithm. The figure to the right illustrates the chosen surveillance routes with the decision-theoretic algorithm. These plots indicate the corresponding room and corridor node numbers, as referenced in Figure 1, that were visited with time.

Nodes Visited (DBN algorithm)	Visit #
R1,R3,R5,R9,R11,R16,R23	1
R6,R8,R17,R18,R19,R21,R24,R26,R28,R30	2
R7,R12,R27,R31	3
R22	4
C1,C6,C10	1
C2,C8	4
C3,C9	3
C5,C7,C11	2

Nodes Visited (Decision-theory algorithm)	Visit #
R2	7
R8	20
R19	20
R22	33
R25	5
R31	6

Table 2: The above tables list the frequencies of each segment of the environment that were visited for the timespan of 447s.

To implement the decision-theoretic algorithm, only the expected costs of each room were taken into account with a look-ahead of $n = 1$ step. After each time instance, the probability of each room will increase to $P_t^{(i)} = 1.0-(1.0-P_0^{(i)})^{t-t_v}$, where $P_0^{(i)}$ is the prior probability of each room and t_v is the time since the room was last visited. When a room is visited, the room's current probability will be reset to its prior probability.

The simulation ran for a total of 447seconds. Results of the simulation are listed in Table 2 and the navigation plots from both algorithms are shown in Figure 2. It is observed that for the entire timespan of the simulation, the decision-theoretic approach planned a route between rooms with the highest expected cost, totaling the number of rooms visited in that timespan to be 91. The DBN planning algorithm, however, planned most of the robot's initial routes to visit corridors with important room neighbours and at a later stage, the algorithm planned routes that visited individual rooms more frequenty. A total of 24 corridors and 46 rooms were visited using the DBN planning algorithm.

Careful observation of the decision-theoretic approach shows that in the total of 91 rooms that were visited, only 6 out of the 31 rooms were actually surveyed. Almost a third of the time was spent revisiting rooms that have high expected costs. The number rooms that were visited using the DBN planning algorithm, however, was 23. The room with the highest expected cost, $R22$, was revisited 4 times compared to the decision-theoretic approach, which planned revisits to the room a total of 33 times.

Overall, the DBN decision algorithm selected paths that were able to survey a large area of the environment. The priority of important regions was apparent as regions with high expected costs were visited more frequently compared to less important regions in that timespan. Instead of surveying individual rooms, corridors with a collection of neighbours with relatively high priors and utility values were visited often to update the beliefs of the collection of neighbouring regions. This eliminates the need to frequently survey individual rooms, which will consequently increase the travelling time of the surveillance robot. Therefore, it is observed from the experimental results that the DBN develops efficient routes that allowed the robot to visit a large area of the environment while maintaining the requirements to survey rooms with high expected costs. This is due to the conditional depedence of the algorithm, which allows the robot to visit only the neighbouring corridors to update the beliefs of the rooms.

6 Future Directions

It is possible to implement the corridor segmentation task as an automated process by finding the convex regions between rooms [Marzouqi and Jarvis, 2004]. This method will further strengthen the spatial relationship between rooms and corridors as it will take into account the visibility of rooms from corridor regions.

The importance of the algorithm presented so far is to establish the beliefs between regions of an environment with the inclusion of spatial and temporal dependencies. Routes that are planned reduces the energy costs of a surveillance robot as it will visit the local neighbours of rooms more frequently, rather than visiting the rooms themselves. However, the algorithm so far does not optimise the planned path between succeeding surveillance goals to include the visitation of intermediate regions - in order to optimally reduce the total expected cost of the surveillance environment. In a simple envrionment, the visitation of intermediate regions is not important as there are only few paths that will lead from one surveillance region to the next. However, in complex and large environments, it may be more advantages for the surveillance robot to visit intermediate areas when transiting between regions. One approach to plan such paths is to model it as a Travelling Salesman problem. By including the expected cost of intermediate regions and possibly the energy costs of a surveillance robot, TSP algorithms could be used to select optimal paths to transit from one surveillance goal to the next.

Another future extension to the decision planning algorithm is to implement it on multiple cooperative robots. In large office buildings, it could be impracticable to utilise a single mobile robot to efficiently survey an entire building. As observed from the simulated results, a single robot is prone to trapping that will cause the robot to only visit areas of the building with high expected costs while ignoring other areas. Even with the implementation of the DBN decision planning algorithm, the number of corridors and rooms a single robot would have to visit in a large office building is insurmountable.

Initial investigations into developing the DBN decision planning algorithm for multiple robots showed that it is feasible to implement such as system on multiple cooperative robots with a decentralised scheme. The market economy strategy for coordinating multiple robots [Dias and Stentz, 2000] proved to be a viable solution to decentralise the coordination of multiple robots. Each robot will be assigned surveillance goals based on the cost of robots to perform tasks. Robots that are able to perform tasks with the least energy cost will be assigned to that surveillance goal. The eventual result is that surveillance goals with high importance will be assigned to robots that are nearest to it. Therefore, an efficient system which reduces the amount of work done by each robot as well as reduces the total expected cost of the system can be achieved.

7 Conclusion

It is concluded that the DBN decision algorithm is able to plan efficient surveillance routes as it takes into account the conditional dependence of neighbouring regions and the temporal dependence of previous observations made. Unlike conventional DBNs, the complexity cost of the proposed decision planning algorithm is managable. Instead of constructing a single DBN to model the entire system, individual DBNs are built to model each neighbouring segments of the system. These DBNs are linked together with the introduction of intraspace nodes, which is used as an approximation of the neighbouring network's current belief.

It is of interest of this author to further extend the DBN decision planning algorithm for implementation on multiple cooperative robots. It is highly inefficient for a single robot to survey an entire building as relevant surveillance events will be less likely observed and the amount of work that has to be done by a single robot is extensive. The design of the DBN decision planning algorithm is highly portable to multiple cooperative robots due to the division of a single DBN to multiple DBNs, which allows the algorithm to be distributed. Therefore, it will be a smooth progression from developing the decision planning algorithm for planning surveillance routes for a single robot to planning the routes for multiple robots.

References

[Dias and Stentz, 2000] M Bernardine Dias and Anthony (Tony) Stentz. A free market architecture for distributed control of a multirobot system. In *6th International Conference on Intelligent Autonomous Systems (IAS-6)*, pages 115–122, July 2000.

[Korb and Nicholson, 2004] Kevin B. Korb and Ann E. Nicholson. *Bayesian Artificial Intelligence*. Chapman & Hall/CRC, 2004.

[Marzouqi and Jarvis, 2004] Mohammed Marzouqi and Ray A. Jarvis. Efficient robotic pursuit of a moving target in a known environment using a novel convex region segmentation. In *2nd International Conference* *on Autonomous Robots and Agents*, pages 13–18, December 2004.

[Massios and Voorbraak, 1998] Nikolas A. Massios and Frans Voorbraak. Issues in surveillance. Technical report, University of Amsterdam, March 1998.

[Massios *et al.*, 2001] Nikos Massios, Leo Dorst, and Frans Voorbraak. A strategy for robot surveillance using the geometrical structure of the environment, 2001.

[Murphy, 1998] Kevin Murphy. A brief introduction to graphical models and bayesian networks. Online, 1998.

[Murphy, 2002] Kevin P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning.* PhD thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, Fall, 2002.

[Voorbraak and Massios, 2001] Frans Voorbraak and Nikolaos A. Massios. Decision-theoretic planning for autonomous robotic surveillance. In *Applied Intelligence*, volume 14, pages 253–262. 2001.