Decision-Theoretic Robotic Surveillance
Massios, N.A.

Citation for published version (APA):
In chapter 4, it was shown that making a hierarchical description of the environment helped produce better surveillance strategies. However, the method presented there was rather \textit{ad hoc} and did not always improve performance.

One of the problems of the hierarchical strategies presented so far is that the expected cost assigned to the abstract nodes does not closely correspond to the actual cost of visiting one. To be more exact, the expected cost of an abstract node is the sum of the expected costs of its children. However, a 5-step path within an abstract node does not always visit all its children and, further, some children are perhaps visited twice. The parameter \( \kappa \) in the revised hierarchical strategy was introduced to deal with some of the side-effects of this inexact expected cost assignment. Despite being an improvement in some cases, this did not produce better results in all test environments.

A much better assignment of the expected costs can be produced if the geometry of the environment is considered. After discussing some general desiderata for clustering an office building, we will concentrate on our specific case of a corridor-based office building. A better method for assigning expected costs to paths visiting abstract nodes is produced with that instance of an office building in mind.

5.1 Clustering desiderata

A \textit{clustering} is defined to be a graph of connected environment locations fulfilling certain criteria. In general, there are many ways in which one would want to cluster an environment. In our case, the main reason for being interested in clustering is the potential for computational time savings. An abstracted \( n \)-step lookahead is prohibitively slow for values of \( n \), large enough to examine every room in our environment before acting. By clustering the environment and approximating the expected cost at higher level nodes, we produce an abstract
description of the problem. Then a decision procedure for planning at the simpler higher levels can be produced. Our clustering is driven by the need to produce such a faster decision procedure that computes approximate solutions.

To generate a clustering, a clustering criterion needs to be selected. There are three main possibilities on what could be used to cluster the environment. It is possible to use the structure present in the type of the indoor environment considered to guide our clustering. A natural clustering of an office building would be to form clusters consisting of separate floors. Within a floor different corridors can form separate clusters and so on. Another possibility is to let the costs guide the clustering. Some environments might have clumps of rooms that are similar, therefore have the same costs in fire presence. A last option is to let the type of permissible paths guide the clustering. For example, it might be reasonable to opt for clusters that produce equal path lengths when visited.

In the case of the surveillance application, the types of paths likely to be useful for a robot performing a surveillance are limited by the properties of the surveillance task. For the purposes of a surveillance task a lot of the possible paths can be ignored because they are not efficient in reducing the expected cost. For example, under mild assumptions, a robot should not stay for many steps in one room before moving to another. Typical paths for a robot performing surveillance are: explore a cluster (visit each room in it at least once), transit through it on its way to something interesting (visit only the rooms needed to get through it), ignore it altogether (do not visit anything) or visit a specific location where, for example, the cost is comparatively high (visit just the rooms on the way to that location and out). These task-dependent path preferences can help us both decide what type of clusters to consider and, related to this, decide to only consider specific paths when assigning expected cost to clusters.

Further, there is an interaction between the shape of the selected clusters and the properties of the considered paths. One such property is that of path length. The path length can be defined to be the total number of room visits along the path. The worst and best case scenarios for exploration path length in a cluster are those where the cluster has a star and a linear topology respectively (fig. 5.1). For the case of a star cluster with \( n \) rooms the exploration length, if we enter and leave at the centre of the cluster, is \( 2n \). For the case of a linear cluster the exploration length is \( n \), if we enter from the sides; for loops this is \( n \), no matter where we enter. This can give us an idea on the bounds of how much time is necessary to explore each type of cluster. For the case of transit paths the situation is rather different. To transit a star can be as short as a single room visit if the entrypoints are in the center of the cluster. To transit a loop can be a lot more complicated with up to \( n/2 \) nodes visited. In fact, for transiting paths the worst case scenario is when the rooms are arranged as a corridor where the path length is \( n \). From the discussion on path lengths it is obvious that when discussing a path within a cluster the entrypoint at which the cluster is entered is relevant.
5.2. **Office-like environments**

As mentioned in the last section, office-like environments contain a lot of structure. Offices are normally organised along corridors. These corridors are in turn connected with each other. If they are at different floors they are connected with staircases or elevators. If they are parallel they are connected at some point along their length. If halls are ignored, it is clear that almost any office-like building can be described using corridor shaped structures.

The building blocks of a corridor are, in turn, star-shaped clusters. By connecting the centres of many star-shaped clusters, a long corridor can be formed. The leaves of those clusters correspond to the offices along the corridor. Further, a hall-shaped structure with many rooms connected to it, can be described as a star-shaped cluster with usually a higher number of leaves than a typical corridor star-shaped cluster.

In general, the four different types of cluster depicted in figure 5.1 are relevant within an office building. Linear clusters correspond to corridors, circular clusters correspond to parallel corridors connected at their ends, stars correspond to the
local structure within corridors and crosses correspond to intersecting corridors. The exact geometrical shape of the rooms and corridors can vary from building to building and largely depends on architectural considerations. However, we believe that the topological structure, although different among buildings, has a common basis.

A clustering process can be developed for office-like buildings that treats level 1 clusters as a special kind of cluster. We call level 1 clusters blocks. The blocks considered have the shape of a star (fig. 5.2). Of course, several other shapes could be considered, but block clusters of this shape are sensible building blocks of corridor-shaped office buildings. The expected cost of star-shaped blocks can be computed directly without examining individual rooms and this simplifies the computation of the expected cost later in this chapter.

A clustering process can be created for office-like environments. This can be based on repeated clustering until the environment cannot be further abstracted. We propose a clustering process with the following properties:

1. (block-based) star-shaped blocks first have to be found to form level 1 of the abstraction.

2. (connected) a path within the cluster should exist between any two nodes in a cluster.

3. (balanced) the resulting tree should be balanced; this means that at each level the exploration paths within the clusters should have more or less equal length.

4. (uniformity) the fire costs and fire starting probabilities have to be uniform within a block.

The last two criteria deserve further discussion. On the issue of tree balancing several options were available. The tree could be balanced on the number of rooms in each cluster (as in section 4.2.1). It could also be balanced on cost so that all clusters of a certain level have equal expected costs. In the next chapter decisions will be taken between different cluster paths. Choosing to balance the tree on path length makes these comparisons fairer. It is the time others are ignored, instead of the particular cost of a cluster itself, that appears more important in our situation.
5.3. Route abstraction simplification

The decision for uniformity in the probabilities and costs within star blocks was taken to simplify the computation of block expected costs. The equations for computing the expected cost of a path in a star rely on the assumption that the fire costs and fire starting probabilities are uniform within a block. It should be mentioned however that this assumption is not very essential. If this assumption was not made, the computation of the expected costs at the block level would have to be more complicated.

If we concentrate on an office-like environment with a single corridor an abstraction tree such as that in figure 5.3 can be created. The focus in the rest of this chapter is on an environment with this type of structure. The equations for the derivation of expected cost assume a corridor-like topology in our building. Of course not all office-like buildings are composed of a single corridor, but this will serve in this thesis to investigate the main issues in surveillance planning.

5.3 Route abstraction simplification

The expected cost assigned to a cluster should depend on the route of subclusters followed within the cluster. A specific cluster can be explored using various different routes and not all routes can be expected to incur the same cost. Ideally, all possible routes within the cluster should be examined. A potentially infinite number of routes exists even within a small cluster if revisits are allowed and it is not computationally feasible to consider all those routes. A way of circumventing this difficulty is to examine a few predefined routes within the cluster. Taking the cluster's entrypoints as a basis, the following possibilities are considered:

- **Exploration routes** $r^e_{X_i}$: routes between entrypoints that visit every sub-cluster.
• **Transit routes** $r^t_{X}$, shortest routes between entrypoints.

• **Ignoring clusters**, all the clusters that are neither explored nor transited are ignored.

While the robot is following a transit or an exploration route, all rooms in the route are sensed and, if necessary, their fires are extinguished. The entrypoint of a route is important in a cluster with more than one entrypoints since then, routes of several directions might exist.

When considering a single room both exploration and transit paths just visit the room itself $r^e_{X_t} = r^t_{X_t} = [X_t]$. In the case of star-shaped blocks a reasonable exploration route visits every node in the star once, and a reasonable transit route visits the central node. This single route property of the blocks simplifies matters because, in a sense, blocks can be treated as rooms. A cluster of level $h > 1$ contains many blocks (or subclusters). In the case of our environment the clusters look like corridors of blocks or subclusters and this simplifies matters. A cluster exploration route is a sequence of subcluster exploration routes that begin at one end and finish at the other end of the corridor. For transit routes the situation is similar.

We give an example of route types for the cases of blocks and clusters.

5.3.1. Example. For the case of block 34 (fig. 5.3) such an exploration route is $r^e_{34} = [r^e_{16}, r^e_{17}, r^e_{16}, r^e_{18}, r^e_{16}, r^e_{19}, r^e_{16}, r^e_{20}, r^e_{16}]$. Similarly only one transit route is possible per block, namely that of visiting the center of the block. For the case of block 66 such a transit route is $r^t_{34} = [r^t_{16}]$.

Since clusters 39, 37 each have a single entrylink, only a single exploration route is possible and in cluster 39 such a route is $r^e_{39} = [r^e_{35}, r^e_{36}, r^e_{35}]$. Routes like $r^e_{39}$ can be recursively rewritten to contain just room routes. Cluster 38 has two entrylinks and so two similar exploration routes are possible one for each direction of exploring. For transit routes the situation is similar. For example, for cluster 28 a transit route is $r^t_{38} = [r^t_{35}, r^t_{34}]$.

### 5.4 Expected cost assignment

Now that a clustering of the environment has been constructed using star-shaped blocks, and standard routes have been assigned to the clusters, we need to assign expected cost specific cluster/route combinations.

To make the equations of this section clearer some shorthand notations are introduced. First, a superscript is used to denote a location's $X$ level $h$ in the abstraction tree. For example, a room will be written as $X^0$ and a block as $X^1$. Secondly, $p_i$ is written instead of $P(f_i \rightarrow 1)$ for the probability of a fire starting at location $X_i$ and $c_i$ instead of $C(f_i)$, for the cost of a fire being present at $X_i$. 
5.4.1 Approximate probability computation

In proposition 3.2.2 the probability $P_t(f_i)$ of the presence of fire at location $X_i$ at a given time since last visit $t$ was defined as:

$$P_t(f_i) = 1 - (1 - p_i)^t$$

where $p_i$ is shorthand for $P(f_i \rightarrow 1)$, the probability of a fire starting during one time-step. The exponential increase of probability $P_t(f_i)$ in time $t$ makes it hard to compute differences of expected costs at different points in time. An approximation of the probability $P_t(f_i)$ is proposed that makes reasoning about the benefit of visiting a cluster easier without affecting the results significantly.

5.4.1 Definition. The approximate probability $\hat{P}_t(f_i)$ of fire presence at time $t$ is defined as the product of the fire starting probability $p_i$ and time $t$:

$$\hat{P}_t(f_i) = p_i t, \quad \text{if } t \ll \frac{1}{p_i}$$

This is not a real probability; the condition of $t \ll \frac{1}{p_i}$ is added to the definition to guarantee that it is not greater than 1. The advantage of this definition over the exact one in proposition 3.2.2 is that $\hat{P}_t(f_i)$ is linear in $t$. We will call any computations made using the approximation of equation 5.1 “linear probability approximations”. The error of the approximation can be determined directly as:

$$E = \frac{(1 - (1 - p_i)^t) - p_i t}{1 - (1 - p_i)^t}$$

The equation for $E$ allows us, given some values for $p_i$ and $t$, to compute the error in our estimation of the probability $P_t(f_i)$. In a concrete example the probability $p_i$ of fires starting would be characteristic of our environment and should be known a priori, while the maximum $t$ would be dependent on the size of the environment.

In order to demonstrate that the quality of the approximation is reasonable a graph of $E$ versus $p_i$ and $t$ was plotted (fig. 5.4). Since our environment has 50 rooms, exploring it would take a maximum of 100 time-steps and so values between 1 and 100 were used for the $t$ axis. In fact, 100 is the worst-case number of steps for the case where the environment is a star, while, in our case, it is a corridor of smaller stars and this makes the exploration path smaller. The probability of fire starting $p_i$ used in the simulations was 0.001 but, in reality, smaller values of $p_i$ would be expected. From the graph it can be observed that the error is always below 5%. So for the environment we are considering, but also for more realistic environments, this approximation is reasonable.
5.4.2 Cost computation

In what follows we give the expected cost computed in the case of rooms, blocks and clusters using the clustering principles of this chapter. Three equations are given for each type of action:

1. $EC_e(X, r^e, T)$ for exploring location $X$, using exploration route $r^e$, given a time $T$ since last visit.

2. $EC_t(X, r^t, T)$ for transiting through location $X$, using transit route $r^t$, given a time $T$ since last visit.

3. $EC_u(X, I, T)$ for ignoring location $X$, for $I$ time-steps, given a time $T$ since last visit.

At time $t$, we need to compute the expected cost of our proposed visiting action (either an exploration or a transit) at a location $X$. Suppose that the last time location $X$ was visited was $T$ time-steps ago, at time $t' = t - T$. Further assume that a visit to location $X$ takes $l$ time-steps where $l$ is the length of the route corresponding to the visiting action taken. Then, the expected cost of this visit to location $X$ is the sum from time $t$ to time $t + l$ of the expected costs for each of the time-steps in the visit of location $X$. These individual costs per time-step depend on the time since last visit $T$. This is seen in figure 5.5. There the
5.4. Expected cost assignment

**Figure 5.5:** The expected costs computed for our actions (area shown in grey).

The x-axis corresponds to time-steps and the y-axis corresponds to cost of location \( X \) per time-step. The shaded area \( a \) is the expected cost \( EC \) the rest of this section computes. Costs between \( t \) and \( t - T \) do not need to be computed, since these are assumed to be costs that are already incurred. Note that we do not talk about \( t \) any longer because the computations we will provide depend just on the time since last visit \( T \) (when the costs were “reset”) and the duration \( l \) of the intended visit.

It was already mentioned that only one of the two types of routes can be followed. The robot can either transit through or explore a cluster while everything else is ignored. The expected cost of the entire environment when a location \( X \) is explored using path \( r^e \) is defined as:

\[
EC = EC_e(X, r^e_X, T_X) + \Sigma_{X' \neq X} EC_n(X', l^e_{X'}, T'_{X}) \tag{5.2}
\]

where \( l^e_{X} \) is the length of the exploration path \( r^e_X \).

If that location is transited instead of explored this becomes:

\[
EC = EC_t(X, r^t_X, T_X) + \Sigma_{X' \neq X} EC_n(X', l^t_{X'}, T'_{X}) \tag{5.3}
\]

where \( l^t_{X} \) is the length of the transit path \( r^t_X \).

These environment expected costs are very important. The robot tries to minimise the expected cost of the entire environment. So what is ignored and for how long is just as important as what is visited.

### 5.4.3 Room expected cost

Focusing on the level of rooms three expected cost equations are again given: one for transiting through, one for exploring and one for ignoring a room. However, the first two are the same because visiting a room or passing through one is the same thing.
When room $X^0$ with a time since last visit of $T_{X^0}$ time-steps is not visited (ignored) for an extra time-step, an approximate expected cost $EC_n(X^0, 1, T_{X^0}) = (p_{X^0}T_{X^0})c_{X^0}$ is associated with it. Generally, if room $X^0$ is not visited for $I$ consecutive time-steps, this incurs an approximate expected cost $EC_n(X^0, I, T_{X^0})$:

$$EC_n(X^0, I, T_{X^0}) = \sum_{t=1}^{I} P_{T_{X^0}, t}(f_{X^0})c_{X^0} \approx \sum_{t=1}^{I} p_{X^0}(T_{X^0} + t)c_{X^0}$$

$$= \frac{I(2T_{X^0} + I + 1)}{2} p_{X^0}c_{X^0} \quad (5.4)$$

where $T_{X^0}$ is the time since last visit of room $X^0$ at the start of the $I$ time-steps.

The exploration route $r^e_{X^0}$ only visits room $X^0$, and so the robot immediately extinguishes any possible fire in that room. If in proposition 3.2.2 $t = t'$, the time-step that the robot visits the room is considered, then the probability of fire presence in that time-step is $P_t(f_{X^0}) = 1 - (1 - P(f_{X^0} \rightarrow 1))^{t' - t'} = 0$. Hence, the expected cost associated with exploring that room is also:

$$EC_e(X^0, r^e_{X^0}, T_{X^0}) = 0 \quad (5.5)$$

Similarly, the expected cost of a route $r^t_{X^0}$ transiting through a room is:

$$EC_t(X^0, r^t_{X^0}, T_{X^0}) = EC_e(X^0, r^e_{X^0}, T_{X^0}) = 0 \quad (5.6)$$

This is also because only room $X^0$ is visited by $r^t_{X^0}$.

It is interesting to give again the expected cost of the entire environment for the level of rooms. We will only do it here at this level because paths at the block or cluster levels can have different lengths. However, as we have mentioned room level transits and explorations are indistinguishable. This makes room-level computation of the environment expected cost fairer.

At each time-step only one room $X^0$ can be explored, or transited through, while every other room $X^0 \in X$ is not visited. The total expected cost at that time-step of the entire environment is:

$$EC = EC_e(X^0, r_{X^0}, T_{X^0}) + \sum_{X^0 \neq X^0} EC_n(X^0, 1, T_{X^0}) \quad (5.7)$$

Because the exploration/transit cost is always 0, this can be simplified to:

$$EC = \sum_{X^0 \neq X^0} (T_{X^0} + 1)p_{X^0}c_{X^0} \quad (5.8)$$

### 5.4.4 Block expected cost

Having described how expected costs should be assigned to rooms, we proceed to discuss the case of star-shaped block clusters. At section 5.3 we have already described how routes are assigned to clusters. As with rooms, three cases are
again considered: ignoring, exploring and transiting through a block. A transit through a star-shaped block only visits its central node, while an exploration visits the central node many times on its way to visiting the leaves of the block (as we have shown in example 5.3.1).

5.4.2. PROPOSITION. (Ignoring after exploration) The expected cost of ignoring a star shaped block $X^1$ of size $s$ for $I$ time-steps is:

$$EC_n(X^1, I, T) = (I(sT + 1 + \sum_{i=0}^{s-1} 2i) + s \sum_{i=0}^{I-1} i)pc$$

$$= (I(sT + 1 + s(s-1)) + \frac{s(I-1)I}{2})pc$$  \hspace{1cm} (5.9)$$

where the size $s$ is defined to be the number of rooms in the block, $T$ is the time since last visit of the block, $p$ is the probability of a fire starting, and $c$ is the expected cost of a fire per time step for each room in the block. The assumption is made that $p$ and $c$ are uniform within the block. It is also assumed that the previous visit was an exploration.

Proof.
We do induction on the size $s$ of the cluster.

**Prove for $s = 1$**
We have, using eq. 5.9, $EC_n(X^1, I, T) = I(T+1) + \frac{I(I-1)}{2}pc = \frac{I(2T+I+1)}{2}pc$. This is equal to the case of a single room (eq. 5.4), thus correct.

**Assume true for $s = k$** Assume that in this case we have: $EC_n(X^1, I, T) = (I(kT + 1 + k(k-1)) + \frac{k(I-1)I}{2})pc$.

**Prove true for $s = k + 1$** In this case the expected cost is that of ignoring the first $k$ rooms (call them $g$) for $I$ steps (the assumed case) + the expected cost of ignoring the $k + 1$ room (call this $g'$) for $I$ steps (see fig. 5.6). In that figure the times since last visit for a star-shaped block are shown. The assumption is that at time $T$ the previous exploration of the block was finished. The times on the leaves in the figure are dependent on the last exploration route followed by the robot. This expected cost can be written as:

$$EC_n(X^1, I, T) = EC_n(g, I, T) + EC_n(g', I, T + 2k - 1)$$
Chapter 5. Path-based Clustering

\[ (I((k+1)T + 1 + k(k + 1)) + \frac{(k+1)(I-1)}{2})pc \]

So equation 5.9 also applies in the \( k + 1 \) case.

5.4.3. Proposition. (Exploring after exploration) The expected cost for exploring a star-shaped block \( X^1 \) of size \( s \) is:

\[ EC_e(X^1, r^*_{X^1}, T) = ((s - 1)^2T + 2(s - 1)^3 + (s - 1)^2 + (s - 1))pc \quad (5.10) \]

where \( s, T, p \) and \( c \) are defined as before and \( r^*_{X^1} \) is understood to be the route of length \( 2s - 1 \) that visits every leaf node in the star once. Again, it is assumed that \( p \) and \( c \) are uniform within the block and that the previous visit was an exploration.

Proof.

Again we provide an induction proof on the size \( s \) of the cluster.

**Prove for** \( s = 1 \) Again this case is equivalent to having a single room. We have \( EC_e(X^1, r^*_{X^1}, T) = 0 \) and this is also what you see in eq. 5.5.

**Assume true for** \( s = k \) Assume that in this case we have:

\[ EC_e(X^1, r^*_{X^1}, T) = ((k - 1)^2T + (2(k - 1)^3 + (k - 1)^2 + (k - 1))pc. \]

**Prove true for** \( s = k + 1 \) We first split the block into the part with \( k \) rooms \( g \) and the new room \( g' \) (see fig. 5.6). Suppose that the exploration taken follows the same order as the last exploration of the cluster. Then we split the exploration of the block into three sections and compute the expected costs in conjunction with those parts (see fig. 5.7). The expected \( EC_1 \) is that of the first time-step of the exploration \( k \) leaf rooms are ignored. The expected cost \( EC_2 \) is that of the second time-step of the exploration and can be seen as ignoring \( g \) while exploring \( g' \). Finally, the last step \( EC_3 \) is that of ignoring \( g' \) while exploring \( g \). Writing this out (ignoring the anyhow uniform \( p, c \) gives:

\[
EC_e(X_k + 1, r^*_{X_k+1}, T) = EC_1 + EC_2 + EC_3 \\
= \sum_{i=1}^{k} (T + 2i) + (EC_e(g, 1, T + 1) - (T + 1)) \\
+ (EC_e(g, r_g, T + 2) + EC_e(g', 2k - 1, 0)) \\
= (kT + k(k + 1)) + (kT + k^2 - T) + \\
((k - 1)^2T + 2(k - 1)^3 + (k - 1)^2 + (k - 1) + (2k - 1)k) \\
= k^2T + 2k^3 + k^2 + k
\]

The equation thus obtained is essentially eq. 5.10 but for \( s = k + 1 \). so eq. 5.10 is proved.
5.4. **Expected cost assignment**

The assumption that the exploration order is the same as that of the last exploration is not essential to the proof. Even if a different order is taken, the differencing in \( T \) in combination with the approximation of the probability gives the same result.

### 5.4.4. Definition. **(Transit after any)** The expected cost of transiting through a block \( X^1 \) of size \( s \) is defined to be the expected cost of ignoring it for the time it takes to transit through it. Since a transit route \( r_{X^1}^t \) for star-shaped blocks only visits the central node of the star, passing through it only takes one time-step. The expected cost of a transit route is then:

\[
EC_t(X^1, r_{X^1}^t, T) = EC_n(X^1, 1, T)
\]

(5.11)

Computing the transit cost by ignoring the block is essentially equivalent to saying that a robot transiting a cluster is moving through it with “its eyes closed”.

### 5.4.5. Definition. **(Exploration and ignoring after transit)** The equations for exploration after exploration (eq. 5.10) and ignoring after exploration (eq. 5.9) are also used for these cases.

We will use equations 5.9, 5.10 even when transits are considered. In those propositions, the assumption was made that the previous visit was an exploration. If that is the case, the expected cost computed by equation 5.10 is a “linear approximation” for \( p \to 0 \). With the introduction of transit routes the state of a block is not necessarily that of figure 5.6 when an exploration begins. In that figure, the times since last visit in each of the nodes are left as if the last visit was an exploration and this figure is used in the proof of “linear approximation” of equations 5.10 and 5.9.

To make everything “linearly approximate”, we would have to consider the interaction between the types of routes at the computation of block expected costs. However, this would yield complicated equations and would evolve many cases based on the type of last visits to a cluster. Since this would be complicated, the decision was made to sacrifice strict “linear approximateness” and to opt instead for a definition of transit where the robot is essentially moving with “its eyes closed”. Although this is still an approximation, it will be shown in the next chapter that the results obtained in this way are very good.

![Figure 5.7: Proof of eq. 5.10.](image)
5.4.5 Higher-level cluster expected cost

Now that the equations for the block case have been defined, we can proceed by defining how the expected cost should be computed for the case of higher-level clusters. Equations 5.12, 5.13 and 5.14 have to be taken to be correct by definition.

5.4.6. DEFINITION. Ignoring a cluster $X^h$ of level $h > 1$ for $I$ time-steps gives an expected cost that can be computed as:

$$EC_n(X^h, I, T) = \sum_{X^{h-1} \in \text{children}(X^h)} EC_n(X^{h-1}, I, T_{X^{h-1}})$$ \hspace{1cm} (5.12)

where $\text{children}(X^h)$ are the subclusters of cluster $X^h$, and $T_{X^{h-1}}$ is the time since last visit for cluster $X^{h-1}$. So ignoring a cluster is the sum of ignoring its subclusters.

5.4.7. DEFINITION. Exploring corridor cluster $X^h$ of level $h > 1$ using a route $r^c_{X^h} = [r^c_{X^h-1}, r^c_{X^h-2}, \ldots, r^c_{X^1}]$ gives an expected cost that can be computed using:

$$EC_c(X^h, r^c_{X^h}, T) = \sum_{i=1}^{n} \sum_{j=1}^{i-1} EC_n(X^{h-1}_i, l_j, T_{X^{h-1}}) + \sum_{k=1}^{j} l_k) +$$

$$EC_c(X^{h-1}_i, r^c_{X^{h-1}_i}, T_{X^{h-1}_i}) + \sum_{k=1}^{i-1} l_k) +$$

$$\sum_{j=i+1}^{n} EC_n(X^{h-1}_i, l_j, \sum_{k=i+1}^{j} l_k))$$ \hspace{1cm} (5.13)

where $T_{X^{h-1}_i}$ is the time since last visit for subcluster $X^{h-1}_i$, $X^{h-1}_i$, $X^{h-1}_i$, $X^{h-1}_i$, $X^{h-1}_i$. \ldots $X^{h-1}_i \in \text{children}(X^h)$ are the subclusters of $X^h$, and $l_k$ is the exploration route length of cluster $X^{h-1}_k$.

In this equation, a cost is added for each subcluster. This corresponds to what happens in each subcluster during an exploration route. A subcluster is first ignored $i-1$ times (while other clusters are explored), then explored, then ignored again $n - i$ times (while other clusters are explored). The position of the cluster in the route is important in deciding how much it is ignored and when. For instance, during the $j$th time the cluster is ignored, it is ignored for $l_j$ room-level time-steps. This is because $l_j$ is the exploration path length of cluster $j$ in the path. This definition makes the assumption that the blocks are organised in a corridor, which is true in the case of the our environment but not in all cases.
5.4.8. Definition. Transiting through cluster $X^h$ of level $h > 1$ using a route $r_{X^h}^t = [r_{X^h_{1-1}}, r_{X^h_{2-1}}, \ldots, r_{X^h_{n-1}}]$ gives a cost that can be computed using:

$EC_t(X^h, r_{X^h}^t, T) = \sum_{i=1}^{n} \left( \sum_{j=1}^{i-1} EC_n(X^h_{i-1}, l_j, T_{X^h_{i-1}} + \sum_{k=1}^{j} l_k) + EC_t(X^h_{i-1}, r_{X^h_{i-1}}, T_{X^h_{i-1}} + \sum_{k=1}^{i-1} l_k) + \sum_{j=i+1}^{n} EC_n(X^h_{i-1}, l_j, \sum_{k=i+1}^{j-1} l_k) \right)$

(5.14)

where $X^h_{1-1}, X^h_{2-1}, \ldots, X^h_{n-1} \in children(X^h)$ are the subclusters of $X^h$, $T_{X^h_{i-1}}$ is the time since last visit of subcluster $X^h_{i-1}$ and $l_k$ is the transit route length of cluster $X^h_{k-1}$.

So the only difference in the expected cost computation between transiting through $X^h$ and exploring $X^h$ is that the central term in the sum $EC_t$ in the case of transits gets replaced by $EC_e$ in the case of explorations.

Knowing the cluster structure, which includes all probabilities, costs and routes, and the equations of sections 5.4.4 and 5.4.5 is enough to compute the expected cost for any cluster/route combination. In fact, the equations about rooms in section 5.4.3 are not necessary, since they are special cases of the equations for blocks for $s = 1$. The definitions of section 5.4.5 treat the block expected cost computation as a closed blackbox computation. This implies that our corridor could consist of blocks of a different shape to stars without any need to modify the equations for computing the expected cost within the corridor.

5.5 Properties of expected cost equations

Several observations can be made about the use of star-blocks to structure the environment graph and the computation of expected cost:

1. For corridor clusters, the entry links used in the visit are important and are considered in the computation of expected cost. Routes visiting the rooms in the cluster in an opposing order produce different costs.

2. The probability of fire starting and the cost of room fire have to be kept uniform at the block level but not necessarily at higher levels. This is not an important limitation. For a different distribution of expected cost the equations of block expected cost would have to become a bit more involved.
3. The block expected cost is computed directly and then accumulation of costs (i.e. summations in eqns. 5.12, 5.13, 5.14) is only used for clusters. This makes expected cost computation slightly faster.

4. The available routes within each cluster are stored together with the cluster. When a route cost needs to be evaluated they can be retrieved from the cluster information.

The expected cost of transiting through, ignoring or exploring a cluster is a linear function its time since last visit $T$. More formally:

**5.5.1. Proposition.** For a cluster $X^h$ of any level $h$, the expected cost of transiting through, ignoring, or exploring a cluster can be written as an equation of the form $EC(X^h, I, T) = aT + b$.

The exact parameters $a$ and $b$ depend on the level of the cluster and its shape, size etc. but the proof of this proposition is relatively easy.

**Proof.**
The proof is by induction. For the case of ignoring a cluster:

**Prove for** $h = 1$ At the block level, eq. 5.9 can also be rewritten to be of the form $aT + b$.

**Assume true for** $h = k$ Assume that the equations for clusters also have the form $aT + b$

**Prove true for** $h = k + 1$ Then for the collection of blocks of level $h = k + 1$ we have using eq. 5.12 that $EC_n(X^k, I, T) = \sum_{X \in \text{children}(X^k)} EC_n(X^k, I, T)$. But then, in turn, this equation is a sum of linear equations and thus also linear.

The proofs for transiting through and exploring a cluster work in the same way and are not reproduced here.

Further, another important observation is that the way in which clusters are decomposed into subclusters is no longer significant. As proof of this statement we consider the simpler possible case of subcluster decomposition in our following proposition. It should be clear that this can be extended to more complicated cluster decompositions.

**5.5.2. Proposition.** Given three clusters $X_1, X_2, X_3$, that eventually comprise cluster $X_5$ (see fig. 5.8). The expected cost computation for exploring, transiting or ignoring cluster $X_5$ is affected neither by introducing an extra cluster $X_4$ in a position between $X_5$ and $X_1, X_2, X_3$ nor by which subclusters are contained within the introduced cluster $X_4$.

**Proof.** For the case of ignoring cluster $X_5$ in the situation of fig. 5.8(a) as:

$$EC_n(X_5, I, T) = EC_n(X_4, I, T) + EC_n(X_3, I, T)$$

$$= EC_n(X_1, I, T) + EC_n(X_2, I, T) + EC_n(X_3, I, T)$$
5.6. Summary

In this chapter we discussed mainly the use of star-shaped blocks in the abstraction. This rather specific shape is at the basis of the description of any office building, so explicit formulas were developed for ignoring, exploring and transiting through star-shaped blocks. If one liked to choose other primitives for a building, introducing a new cluster shape would only affect the computation of the expected cost at the block level since new equations have to be introduced for the new shape. The accumulation of expected costs at the cluster level should remain the same provided, of course, that the case of a corridor environment is considered.

The linearity property makes comparison of expected costs and predictions of their evolution simpler. It is a property which can simplify the process of decision-making between clusters. We also proved a property on the interaction between the abstraction hierarchy and the expected cost assignment which states that the assignment is independent of the exact ordering of the clusters over the
block level. This suggests robustness of the proposed method against arbitrary decisions on the position of cluster boundaries and this is something we will confirm in the next chapter.