Plausible Environment Reconstruction Using Bayesian Networks

Robert G. Price and Scott D. Goodwin

School of Computer Science
University of Windsor
Windsor, Ontario, Canada
{price9|sgoodwin}@uwindsor.ca

Abstract

To save precious time and space, many games and simulations use static terrain and fixed (or random) reconstruction of areas that a player leaves and later revisits. This can result in noticeable differences between the reconstructed area and the player's recollections (or expectations). These differences can lessen a player's immersion in the game, or the usefulness of the simulation. We propose an approach for environment reconstruction that uses a Bayesian Network to quickly and easily calculate likely effects that external factors have on the environment. The reconstruction of revisited areas becomes less disconcerting and permits the incorporation of plausible changes based on unobserved, yet reasonably expected, events that could have occurred during the player's absence.

Introduction

The motivation behind this project is to have a "Level-of-Detail Artificial Intelligence" (LODAI) (Brockington 2002) approach to determine what type of terrain to draw in a large simulation or role-playing game. Many games allocate no CPU time for events unfolding outside a player's view. For example, if you break windows in the game Fable (Lionhead Studios 2004), and then leave the zone and come right back, the windows will all be replaced. If you were to stay in the area and watch, the windows will not get fixed. These games can have the result that if the player is watching, one thing happens, and if the player is not watching, a different set of events occur. This disconnect can be undesirable in games and simulations. A LODAI approach might use a small amount of CPU time to update the world that is outside the player's view periodically. With such an approach, if the player came back into the area immediately, there would be no change (all the windows would still be broken). After a long period of time, everything might be back to normal (all the windows would be replaced). In an intermediate point in time, a plausible intermediate state could be represented (after a few days, the windows on the expensive homes could be replaced, while the other windows would still be broken, or boarded up).

Bayesian Networks

A Bayesian Network (BN) is a data structure that can be used to represent dependencies among variables and to give a concise specification of a joint probability distribution. A BN is a directed acyclic graph with the variables represented as nodes in the graph, with an edge between a parent node and a child node if the parent node directly influences the child node. It is usually easy for a domain expert to decide what influences exist in the domain. A BN representation can take much less space than a joint probability distribution to specify, and also be easier to represent. A good introduction to Bayesian networks can be found in (Charniak 1991). A Dynamic Bayesian Network (DBN) is a BN that represents a temporal probability model. A DBN can be used model the environment over many time slices. At each time slice, the variables have parent nodes that are in its own time slice, or in the time immediately preceding it. We discuss our BN later in this paper.

We chose to use a Bayesian Network to reconstruct the environment since it is easy to model many independent variables efficiently, and it is quick to calculate the desired probabilities given the available information.

Using A BN to Reconstruct the Environment

We used the Microsoft Bayesian Network Editor (Microsoft Research 2001) to model the Bayesian Network, which can subsequently be used to determine what type of terrain to draw when a player walks into an area that he or she has not visited for a while. Several nodes could affect the terrain that would be ultimately drawn. The weather, the vegetation, human activity in the area, etc. can all have an effect on what to draw. Some of these nodes could also have effect on other nodes: a drought (rainfall) could lead to water being scarce (water availability), which could affect a village (human presence).

As a simple example, a game could have a BN that has rainfall and temperature as parent nodes to determine what to draw. In a given area under normal conditions, the ground can be covered with healthy green grass and a healthy tree could be drawn. In times of little rainfall and high temperatures, the grass could be drawn using a yellow color, and instead of using a 'healthy-tree' model, an 'unhealthy-tree' model could be used. In a very cold winter with high precipitation, snow can be covering the ground instead of grass, and a model of the tree with snow on the branches can be used. It would make it simple to integrate an approach like this into a game, since there would be little chance of negative interactions with other parts of the game. For example, if the non-player characters in a game use waypoints to find their way around, the same waypoints could be used to navigate around the tree, regardless of what model of the tree the player sees.

This approach does not have to be limited to 'realistic' reconstruction of the environment. In the game Fable, your character's appearance changes based on your actions – a very evil person will have horns growing out of his head. There is nothing stopping a game that uses a BN approach to give clues to the player through the environment. For example, if there is an evil wizard living in a tower, the vegetation in the surrounding area could appear sick and dying. Once the evil wizard has been driven away, the plants can appear healthy and vibrant again.

This approach could also take into account the actions of the characters in the game when determining the environment to draw. If there was a brawl in a tavern recently, a player who wanders into the room might see evidence of the recent fighting (broken glass and furniture). After a period of time, however, the tavern would go back to normal, and there would be little evidence for the player to see that there was a fight. If there was a battle recently between a mad wizard who likes to cast fireballs and a bunch of goblins, a player wandering into the area would see that the ground is blackened, and that there are some corpses on the ground.

Difficulties with this approach

Implementing this kind of solution does come with additional costs. There is a knowledge engineering cost with using this kind of approach. Coming up with the structure of the network can be relatively easy by examining the effects various factors have on each other. However, it might take a fair bit of tuning to get realistic or reasonable probabilities. It would detract from a game if the BN that was used creates an environment that varies greatly from what the character expects. Testing must be done with the BN to ensure that a small drought doesn't turn an ancient forest into a dessert. The values that we assigned to the nodes in our BN were generated by hand. Depending on how complex you make your network, it might take a considerable amount of time to enter in all of the values and test them. The good news is, once you have acceptable results for one area, it should be easy to reuse your BN in similar areas with little modification. An area in a game that should be desert will require a different BN than an area that should be grasslands, but once you have a BN that gives you suitable results for one grasslands area, you should be able to use it in grassland areas throughout your game.

In addition to any difficulties in getting your BN to produce acceptable outputs, there will be additional work for the game's artists to come up with additional models. Instead of a single tree for a certain location in the game, several trees would need to be made, and the game engine would pick which one the player would see, based on the BN. In addition to the extra time and effort to come up with the models, there would be additional storage requirements to consider when installing the game. This might not be a very large cost, since many models can likely be reused in different areas of the game. For example, if an artist has to make several different models for each tree in a certain area, those models can also be used for the trees in different areas.

There might also be additional costs and other difficulties in keeping track of the player's actions. In a multiplayer game, several players would be able to simultaneously affect the environment. It could be difficult to store all of the data, so that each player can revisit an area at a later date to see their effect on it. However, the data shouldn't need to be stored for very long. If a player is involved in a big battle on a certain day, the details of the area can be aggregated when a certain time passes after the player leaves the area. For example, after a few days, the exact placement of the bodies can be discarded, and only the number and the relative area can be kept. If the player returns after a week, he or she will likely remember that there was a battle in the area, but not the exact placement of the bodies. Even if the player notices that the bodies are in a slightly different orientation, it might be explained away by something else (perhaps someone coming by to search for treasure).

In a single-player game, it might be possible for several years to go by in the game before a player returns to an area. The BN should be able to relatively easily recreate an environment that the player believes is realistic. In a massively multiplayer game, it is unlikely that an area will remain unvisited for that long of a time. In the worst case, a certain area might constantly have one player-character or another in the area. In this situation, other approaches will likely have to be taken, as using a BN to update the environment would be inappropriate (since it would be disturbing to the players if the environment they are in suddenly changed for no apparent reason).

Future Work

This method for generating terrain could benefit from applying any techniques that simplify the calculations in any Bayesian Network. For example, we could explore using only singly connected networks, or polytrees. A method for joining nodes in a Bayesian network into a tree of clusters is found in (Huang and Darwiche, 1996). There also might be a benefit to using a simpler representation of the Bayesian Network. For example, if the node "fire", which represents the probability of a forest fire occurring, has two parent nodes: "rainfall" and "temperature", which represent the amount of recent rainfall and the current temperature, it might be advantageous to have a representation where we can specify the probability of a large forest fire starting is zero if there was very high amounts of rainfall recently regardless of temperature, rather than having to specify the probability for each temperature. Advances in Bayesian Network technology are continuously appearing in the literature, which will inevitably inspire improvements in our proposed approach.

This LODAI approach could be used in a simulation, such as the Virtual Archaeologist Project (Kobti and Goodwin, 2005), which is intended as a simulation of the settlement and farming practices of the Pueblo Indians of the Central Mesa Verde region of Southwest Colorado between A.D. 600 and 1300 (see Crow Canyon, 2004). The simulation uses data from the archaeological site, soil data, tree-ring data, etc. An educational game is being developed for an on-site archaeological learning center for visitors to explore aspects of the population, such as settlements, artifacts, architecture, farming, trading, hunting, and raiding. The user will be able to view (and affect) an area, and then jump ahead several years and view the area again to see what changes occurred. For example, if the user could move the simulation forward 50 years to see what effects settlement and erosion have on the terrain. Our Bayesian Network based environment reconstruction could be used to determine plausible changes in the environment. If there were a drought for several years, then the plants in the area would have little growth for those years. If a forest fire happened in an area 5 years ago, then there would be few, if any, large trees. A Bayesian Network can combine many factors to determine the most likely consequences.

The DBN that we created for this simulation has the following nodes: Rainfall, Temperature, Water Availability, Fire, War, Vegetation, Wildlife Presence, Human Presence, and Settlements. The nodes at year *i* can be affected by the nodes at year *i-1* (Vegetation in an area one year helps determine what will be there during the next year). Additionally, Water Availability has Rainfall as a parent node. Rainfall, Temperature, and War are the parent nodes of Fire. Fire, Water Availability, and Temperature are the parent nodes of Vegetation. Vegetation and Water Availability are the parent nodes of Wildlife Presence. War, Fire, Water Availability, Vegetation, and Wildlife Presence are the parent nodes of Human Presence. Human Presence and Fire are the parent nodes of Settlements. Rainfall, Temperature, and War have no parent nodes in the same year. The complete BN that was used for this example can be found in (Price and Goodwin, 2006).

Conclusions

This paper has demonstrated that it is possible and quite simple to use a Bayesian Network to help determine what type of terrain to render in a simulation or game. This could be used to make games and simulations more realistic than their counterparts that simply use static terrain. The contribution of this paper is to demonstrate that determining what type of terrain to draw can be done quickly and easily, and can add to the realism in a simulation, while not taking up much CPU time. The importance of this LODAI approach using minimal CPU time is that it might give the player the impression that the simulation has modeled the world more accurately than it actually has, and that the simulation is doing more work than it actually is. For a relatively small cost, the believability of the game or simulation can be immensely improved.

Acknowledgements

The authors of this paper would like to thank NSERC. The first author is supported by a PGSD, and NSERC Discovery Grant 122131-04 supports the second author.

References

Brockington, M. 2002. Level-Of-Detail AI for a Large Role-Playing Game. *AI Game Programming Wisdom*, pp. 419-425, Charles River Media.

Charniak, E. 1991. Bayesian Networks without Tears. *AI Magazine*, 12 (4): 50–63.

Crow Canyon Archaeological Center. 2004. (Learning Center) http://www.crowcanyon.org/learning.html

Huang, C., and Darwiche, A. 1996. Inference in Belief Networks: A Procedural Guide. *International Journal of Approximate Reasoning*, 15 (3): 225–263.

Kobti, Z., and Goodwin, S. D. 2005. Virtual Archaeologist http://cider.cs.uwindsor.ca/Archive/Village3D.ppt

Lionhead Studios. 2004. *Fable*. Microsoft Game Studios.

Microsoft Research. 2001. *Microsoft Bayesian Network Editor and Tool Kit*. http://research.microsoft.com/adapt/MSBNx/

Niedermayer, D. An Introduction to Bayesian Networks and their Contemporary Applications, http://www.niedermayer.ca/papers/bayesian/bayes.html

Pearl, J. 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. *Morgan-Kaufmann Series in Representation and Reasoning Archive*. Morgan Kaufmann Publishers Inc.

Price, R. G., and Goodwin, S. D. 2006. *Reconstructing Terrain using Bayesian Networks*, 6 pages, Unpublished Manuscript.

Russell, S. J., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*, 2nd edition, Pearson Education Inc, 137–158, and 559–568.