What's Going on? Multi-sense Attention for Virtual Agents

Tim Balint and Jan M. Allbeck

Laboratory for Games and Intelligent Agents
George Mason University
4400 University Drive, MSN 4A5
Fairfax, VA 22030
{jbalint2,jallbeck}@gmu.edu

Abstract. When designing virtual humans, it is imperative that the virtual agent behave in a human-like fashion. In certain circumstances, this requires agents to be bounded in their ability to sense and understand the environment. To this end, we create a methodology that provides perceptual attention based on a linear combination of senses. We use different heuristics on the object and events that can be sensed by a virtual agent, and combine these scores to create an overall score for a given object or event. This allows the virtual agent to perceive interesting or unexpected items. To demonstrate our technique, we give an example showing the ability of using a linear combination of senses.

1 Introduction

Virtual humans play a growing role in movies, games, and simulations. The primary role of these agents is to behave like physical humans, requiring the agent to make human-like decisions. As physical humans interact within the bounds and understanding of their environment, which can change unexpectedly, virtual humans should be able to as well. To simulate this, an agent must be able to know and understand what is going on in their environment, as can be seen from Figure 1.

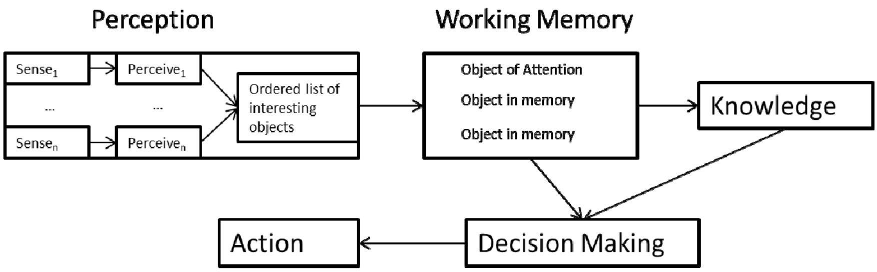

Fig. 1. A sample agent framework, with the perception and working memory portions expanded. In this figure, the perception module is broken down to display a multitude of sensing abilities.

R. Aylett et al. (Eds.): IVA 2013, LNAI 8108, pp. 349–357, 2013.
© Springer-Verlag Berlin Heidelberg 2013

An agent interacting within a semantic environment, also known as a smart object environment [1], must be able to use and understand the semantics found within that environment. Complex computer vision systems have been designed to imbue an agent with synthetic vision, allowing the agent to sense their environment [2]. However, data in a virtual semantic environment is attached to the objects; this can make understanding the virtual world as simple as looking up values in a table. If the agent's model of perception consists solely of looking up values, then the agent exists inside a fully known environment. This may not be plausible for certain cognitive processes (such as agent memory models and decision-making processes), but can be addressed by modeling various aspects of agent perception. Also, when simulating virtual humans, it is natural to believe that the five human senses will suffice. However, humans augment their senses every day, with tools as simple as thermometers or as complex as radar and night vision systems. Instead of designing all possible senses, another alternative is to generalize what a percept is, and from there, add specifics based on the requirements of a simulation. Determining the best solution for such a problem is generally context dependent, as the information present within the environment plays a necessary role in how the agent can process its environment. A simulation author facilitates this process by creating the environment and populating it with information. Therefore, the simulation author is able to choose how an agent understands its world.

An agent can sense the environment, but if the environment is sufficiently complex, there could be too much information for the agent to realistically process. Physical humans do not process all objects within an environment at once, and generally only concentrate on a few at a time [2]. For example, a person who glances at a cluttered coffee table may not notice their keys amidst the remotes and books. In order to create more believable virtual humans, an agent may need to prune away at its sensed search space, keeping only the interesting or stimulating information- thus adding the realistic effect of being unobservant or overlooking.

To allow virtual humans to understand their environment, we propose and implement a generalized framework for agent sensing and perception. This framework includes:

- A generalized agent sensing system that allows simulation authors to create specific senses for a given simulation.
- A perception system that uses a heuristics to create sense attention through a bottom-up and top-down process.
- A linear combination of forms of perception that allow for agents to combine perception scores across multiple senses.

2 Related Work

Perception for virtual agents is a diverse topic, as it is a vital part of the sense-think-act cycle found in both physical and virtual modern agents. A large sector of the computer vision community has focused on agent perception, and a survey can be found in [3]. Many of these systems create false color images or saliency maps, the latter being especially used in the computer vision community [4]. Specifically within the virtual agents community, work has been made in top-down perception [5], bottom-up perception [6], and a combination of the two [7]. All of these systems provide agents with a sense of vision while [6] and [7] also provide an auditory sense. They forgo much of the processing that is done in the vision community, as all the information available to them can be accomplished in a search of the object. We have adopted this format, and much like these systems, use information inherent and available to the agents. We chose to generalize the sensing ability to allow for the definition of multiple senses, instead of specific formats for each individual sense.

Many systems couple visual attention with perception, which allows an observer to determine what the agent actually perceives. The agent could give attention to events in their environment [7], or other agents [8]. Visual attention has also been used to facilitate memory systems [9] and general agent decision-making [5]. While some of these systems provide multiple senses for the agent to receive information though, they handle information from each distinct sense separately, that is, there is no attempt to combine senses. We have chosen to allow agents to combine different senses within their perception system when computing costs of visual attention, making our system determine sensory attention, rather than just visual attention.

Perception and visual attention have been added to many different systems for the purpose of creating more intelligent virtual agents. Embodied conversational agents in particular have used perception to allow them to understand the physical or virtual human they are conversing with [10]. This has been researched enough to warrant study into a markup language for perception in conversational agents [11]. There has also been an emphasis on creating perception for embodied agents interacting with a virtual environment, and many of the systems listed above do so including [5] [6] [7]. Reactive agents, such as those found in [12] also use perception and visual attention to understand concepts of a virtual environment. [13] and [14] take this one step further by using visual attention as a pre-cursor to other motor activities, such as reaching, searching, and catching. We focus on perception for agents within a virtual environment. Our agents use an understanding generated through a combination of multiple senses to create an attention span and populate the working memory component seen in Figure 1. This means that the agent focuses mostly on objects within the environment, and so our work is not necessarily suited for use in agent conversation.

While there has been a great deal of work on creating multiple sense attention for virtual agents, many of the solutions focus only on information from one sense at a time. These models process information as it is received, and most perform (visual) attention on a set of objects once it is seen in the environment, making them an event driven model of perception. However, when attempting to combine information from multiple senses and reason over these senses, processing objects one at a time is no longer manageable, especially when the reasoning for one object requires understanding the surrounding environment. In an attempt to combine multiple senses, we provide a method of grouping, reasoning, and combining information about objects, in order to create a sense attention.

3 Methodology

3.1 Object and Action Representations

The representation of objects in our environment are inspired by Smart Objects [1], which contain sets of properties representing semantic information. Semantic properties, hereby denoted p, are categorized into sets, S, and these sets make up the group of semantic information. Objects may contain one property per set, and may be marked with many different sets. For example, properties p_i and p_j are members of set S_m, and property p_k is a member of set S_n. An object may have either property p_i or p_j, without an effect on its ability to also have p_k. These semantic sets are generally contained globally, and individual objects may inherit a single property from these sets, although the inherited property may change during the simulation. Before the simulation, the sets of semantic properties can be authored by a user, or possibly determined from a common-sense database [15]. A subset of the property sets we are currently using is found in Table 1.

Table 1. A subset of the property sets understood by our perception system

Property	**Type**
Olfactory type	String set
Visual Hue/saturation/brightness	Integer set
Visual Luminance	Integer set
Auditory frequency/Intensity	Integer set

3.2 Sense Preprocessing

In order to endow virtual agents with sense attention, the agent must be able to determine if it can sense an object. We define sensing as the ability an agent uses to determine the presence of a semantically labeled object through some semantic information attached to that object. We represent a change in semantic information upon an object as an event. For example, an agent can sense a pizza through seeing its shape and color or through a pizza's distinct smell. Different senses should discern different properties of a semantic object, although some properties can manifest themselves in many ways. A semantic object that is hot could have a *heat_signature* semantic information type, detectable by an infrared sense or touch sense. An object that is hot could also glow red and therefore be detected by a visual sense. We consider these two semantic properties distinct, and each one is added to the virtual environment separately.

We provide our agents a method to determine which objects are in their general sensing area. Much like [6], our agents have an area they can sense in, which is determined by the simulation author, and generally varies by sense and by agent. This is determined by two variables, a subtended angle α and a sensing distance d. α and d form polar coordinates that allow agents to determine what objects are within their sensing area, as seen in Figure 2. Unlike techniques such as ray-tracing, our technique does not inform an agent if an object is blocked by another object. While this removes some realism from certain sensing abilities, we believe that it does not negatively impact the agent. Certain senses, such as auditory and olfactory senses, do not rely on objects being in the agent's line of sight, but being within a certain range of the agent, thus providing them with back up receptors to visibility.

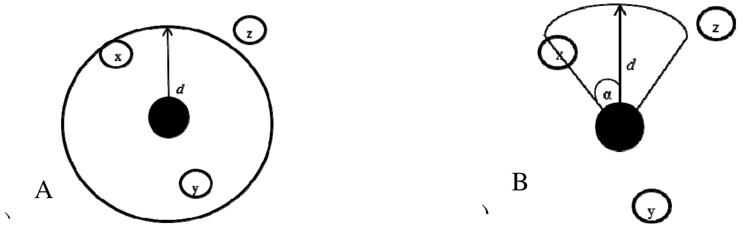

Fig. 2. Sense pre-processing areas A) α is 360 degrees around the agent. In this example, x and y are sensed by the agent, but z is not. B) α is less than 360 degrees. In this example, only object x is sensed.

To determine if an object is in the sensing area, we keep a list of all sorted objects that are then examined by the agent. Objects that exist within this sensing area are fed into the sense where they are pruned from the agent's consideration based upon the semantic properties they contain. At this stage of processing, it is only important for the agent to determine if an object has one semantic property useful to the sense. For example, certain properties, such as a frequency and decibel level, are generally associated with an auditory sense. If an agent is examining an object that has semantic information from one of these two properties, the object is still passed on for further processing. This allows for some generalization between sets and between objects.

3.3 Perception Attention

Physical humans can only process a limited number of items at a time, which is simulated by providing the agent with an upper bound on the working memory portion of the agent system (See Figure 1). Only unusual or important information is generally retained, thus rejecting much of what a physical human senses. However, what is generally considered interesting by one sense, when experienced by multiple senses, may not be considered as interesting. In order to create more plausible virtual humans, we believe that this ability should also be mimicked. To accomplish this, we model virtual human perception over a multitude of senses.

In order to create specificity when designing perception for multiple senses, we employ a series of heuristics, similar to [12], a subset of which is seen in Table 2. However, unlike [12], which determines common attributes from sensed objects, we attempt to rank objects based on pre-authored information readily available from either the agent or the environment (such as the distance between objects or an object's hue). We design heuristics based on two forms of comparison: agent-object interaction, which we label as top-down interactions, and object-object, or bottom up interactions.

Table 2. A sample of the heuristics used in our perception system. *Sense* is the sense type we designed the heuristic for. *Form* is whether the heuristic compares objects to other objects (Bottom-up) or objects to an agent (Top-down). The *type* of heuristic shows the basic way in which it is calculated.

Name	Sense	Form	Type
Auditory Saliency	Auditory	Bottom-up	Comparative
Olfactory Saliency	Olfactory	Bottom-up	Comparative
Velocity Saliency	Multiple	Bottom-up	Comparative
Interacting	Multiple	Top-down	Selective
Using	Multiple	Top-down	Selective
Useful Object	Multiple	Top-down	Selective

Top-down interaction heuristics allow an agent to create a personal score with the object, therefore yielding different results for different agents. For example, an *interacting* heuristic, determines if an object (such as another agent) is interacting with the agent. From Table 2, it can be seen that we regard all top-down heuristics as selective heuristics, which follow the basic form found in Equation 1.

$$h(object, agent) = \begin{cases} score & if\ object\ passes\ selection\ test \\ 0 & otherwise \end{cases} \tag{1}$$

Other heuristics, which employ object-object interactions, are considered bottom-up heuristics. As can be seen from Table 2, many of these heuristics use saliency and are comparative. Unlike most systems, we do not use computer vision techniques to create a saliency image, but instead perform comparisons based on the objects through the sense pre-processing phase. While in some cases this will remove information that is generally used in saliency maps, it maintains object-object interaction while removing certain forms of pre-processing (such as background subtraction). As can be seen from Table 2, many bottom-up heuristics are comparative, and so the object must be compared to all other objects within the environment. These heuristics, being comparative in nature, also create relative perception within the heuristic. A less intense smell, when compared to a stronger one, will be scored much lower using these types of heuristics.

Many heuristics used by the agent require statistical processing. Heuristics such as saliency require not only comparisons to each object in the area, but also to the average object score. We implement comparisons to the average, maximum, and minimum score for a set of objects for a given heuristic. Since heuristics can be re-used over multiple senses, it is not efficient to embed these statistics within the heuristics, but better to process them after the heuristic has been run on all objects.

After a heuristic, H, is processed for all objects, it is normalized with a weight, *ws*, over the total of all weights, *wn*, and added to a hash table object->score. After all heuristics are processed for a given sense, that sense is normalized with a weight *w* and total of all weights *n* as well. The total score for a given object over all senses is then given as a summation, seen in equation 2. By using linear weights on both heuristic scores and senses, a simulation author can control whether one heuristic or sense should dominate the others. Finally, the hash table is pruned to only the objects that score highest from senses, creating a bounded memory in complex scenes, such as scenes found in [16].

$$u = \sum_{i=1}^{t} \frac{w_i}{n} \sum_{j=1}^{s} \frac{ws_j}{wn} H_j(o) \tag{2}$$

4 Analysis and Results

4.1 An Example

In order to highlight the difference between using an agent-based linear combination perception system and one using simple selection, a sample scenario has been created. We have modeled a complex diner environment, which contains several objects. Situated within this diner is a kitchen, complete with a stove, microwave, pots, pans, cups, food, and other objects typically found in a kitchen. Each of these objects has been labeled with different semantic properties, a subset of which is found in Table 1. Certain properties from these sets (such as sound properties) are also added to objects through events. Actuators for the virtual agents are provided through Smartbody [17]. Additional examples can be found with the accompanying video.

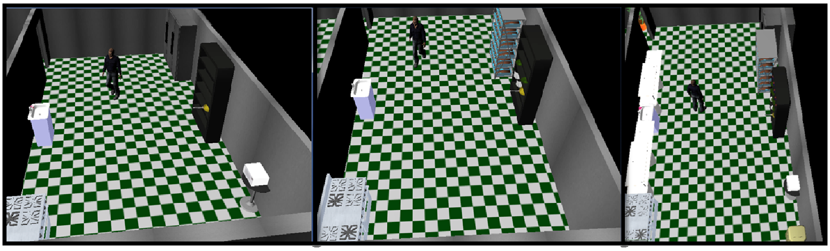

Fig. 3. The agent observing its environment. Objects seen in the environment are the objects perceived by the agent. Left: The agent observes its environment using an attention selection method. Middle: The agent observes its environment using our linear combination method. Right: The full kitchen environment.

Our scenario has an agent enter the kitchen, examining its surroundings and reporting on the objects it perceives, using visual, auditory, and olfactory senses. All items have at least visual semantic information, and many items, such as the refrigerator or oven, have a sound or smell associated with them. Figure 3 displays the agent's observation using a selection method on the left and a linear combination method in the middle. As can be seen from the images, several of the objects are the same. However, some items, such as the refrigerator, are ignored by the linear combination method. As the refrigerator is only recognized by the auditory sense with both methods, the uninteresting sound that it makes goes unnoticed when more interesting objects, such as the glasses and cups, are in range. The refrigerator's sound is noticed with the selection method due to the lack of objects that have auditory semantics, and that the auditory sense is the last sense to be checked.

5 Conclusions and Future Work

We have provided a method to create perceptual attention based on a combination of different senses. This is accomplished using a preprocessing step to determine objects and events capable of being considered by a given sense, and a ranking step to determine objects that are useful or interesting to the agent. This provides the agent with an ability to differentiate objects in its environment through the use of all available information.

Future research will examine the heuristics used in this work. Certain heuristics may be applicable only for certain situations, and so the ability for the agent to adapt and control its heuristics may provide faster and more interesting results. Optimizations on these heuristics, especially the comparative ones, would also prove useful.

References

1. Kallmann, M., Thalmann, D.: Modeling Objects for Interaction Tasks. In: Eurographics Workshop on Animation and Simulation, Lisbon, pp. 76–86 (1998)
2. Hill, R.W.: Perception Attention in Virtual Humans: Towards Realistic and Believable Gaze Behaviors. In: AAAI Fall Symp. Simulating Human Agents, pp. 46–52 (2000)
3. Peters, C., Castellano, G., Rehm, M., et al.: Fundimentals of Agent Perception and Attention Modeling. In: Emotion-Oriented Systems. Springer, Heidelberg (2011)
4. Canosa, R.: Modeling Selective Perception of Complex, Natural Scenes. International Journal on Artificial Intelligence 14, 233–260 (2005)
5. Joost van, O., Frank, D.: Scalable Perception for BDI-Agents Embodied in Virtual Environments. In: International Conferences on Web Intelligence and Intelligent Agent Technology, pp. 46–53 (2011)
6. Herrero, P., Greenhalgh, C., De Antonio, A.: Modeling the Sensory Abilities of Intelligent Virtual Agents. In: Autonomous Agents and Multi-Agent Systems, pp. 361–385 (2005)
7. Kim, Y.-J., van Velsen, M., Hill Jr., R.W.: Modeling Dynamic Perception Attention in Complex Virtual Environments. In: Panayiotopoulos, T., Gratch, J., Aylett, R.S., Ballin, D., Olivier, P., Rist, T. (eds.) IVA 2005. LNCS (LNAI), vol. 3661, pp. 266–277. Springer, Heidelberg (2005)
8. Rymill, S.J., Dodgson, N.A.: Psychologically-Based Vision and Attention for the Simulation of Human Behavior. In: Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia (GRAPHITE 2005), pp. 229–236 (2005)
9. Cha, M., Cho, K., Um, K.: Design of Memory Architecture for Autonomous Virtual Characters using Visual Attention and Quad-Graph. In: ICIS, pp. 691–696 (2009)
10. Peters, C.: Direction of Attention Perception for Conversation Initiation in Virtual Environments. In: Panayiotopoulos, T., Gratch, J., Aylett, R.S., Ballin, D., Olivier, P., Rist, T. (eds.) IVA 2005. LNCS (LNAI), vol. 3661, pp. 215–228. Springer, Heidelberg (2005)
11. Scherer, S., Marsella, S., Stratou, G., Xu, Y., Morbini, F., Egan, A., Rizzo, A(S.), Morency, L.-P.: Perception Markup Language: Towards a Standardized Representation of Perceived Nonverbal Behaviors. In: Nakano, Y., Neff, M., Paiva, A., Walker, M. (eds.) IVA 2012. LNCS, vol. 7502, pp. 455–463. Springer, Heidelberg (2012)

12. Steel, T., Kuiper, D., Wenkstern, R.Z.: Context- Aware Virtual Agents in Open Environments. In: Sixth International Conference on Autonomic and Autonomous Systems, pp. 90–96 (2010)
13. Chopra-Khullar, S., Badler, N.I.: Where to Look? Automating Attending Behaviors of Virtual Human Characters. In: Autonomous Agents, Seattle, pp. 16–23 (1999)
14. Yeo, S.H., Lesmana, M., Neog, D.R., Pai, D.K.: EyeCatch: Simulating Visuomotor Coordination for Object Interception. In: SIGGRAPH, p. 4 (2012)
15. Li, W., Allbeck, J.M.: Virtual humans: Evolving with common sense. In: Kallmann, M., Bekris, K. (eds.) MIG 2012. LNCS, vol. 7660, pp. 182–193. Springer, Heidelberg (2012)
16. Hill, R.W., Kim, Y., Gratch, J.: Anticipating Where to Look: Predicting the Movements of Mobile Agents in Complex Terrain. In: Autonomous Agents and Multi Agent Systems, Bologna, pp. 821–827. ACM (2002)
17. Feng, A., Huang, Y., Xu, Y., Shapiro, A.: Automating the Transfer of a Generic Set of Behaviors onto a Virtual Character. In: Kallmann, M., Bekris, K. (eds.) MIG 2012. LNCS, vol. 7660, pp. 134–145. Springer, Heidelberg (2012)