

# 4

## Vision Zones and Object Identification Certainty

*Steve Rabin*

4.1	Introduction	4.4	Producing Appropriate Behavior from Identification Certainty
4.2	Vision Zones and Object Identification Certainty	4.5	Conclusion
4.3	Incorporating Movement and Camouflage		References

### 4.1 Introduction

Within games it is common for human AI agents to have a very simplistic vision model. This model usually consists of three vision checks performed in this order (for efficiency reasons): distance, field of view, and ray cast. The distance check restricts how far an agent can see, the field-of-view check ensures that the agent can only see in the direction of their eyes (not behind them), and the ray cast prevents the agent from seeing through walls. These reasonable and straightforward tests culminate in a Boolean result. If all three checks pass, then the agent can see the object in question. If any of the checks fail, the agent cannot see the object.

Unfortunately, this simplistic vision model leaves a very noticeable flaw in agent behavior. Players quickly notice that an agent either sees them or not. There is no gray area. The player can be 10.001 meters away from the agent and not be seen, but if the player moves 1 millimeter closer, the agent all of a sudden sees them and reacts. Not only is this behavior odd and unrealistic, but it is easy to exploit and essentially becomes part of the gameplay, whether intentional or not.

The solution to this problem is twofold. First, we eliminate the Boolean vision model and introduce *vision zones* coupled with a new concept of *object identification certainty* [Rabin 08]. Second, we must have the agent use the object identification certainty to produce more nuanced behavior.

---

## 4.2 Vision Zones and Object Identification Certainty

Human vision is not binary, which is the core problem with the simple vision model. There are areas where our vision is great and areas where our vision is not so great. Our vision is best directly in front of us and degrades with distance or toward our periphery. This can be approximated by using several vision zones, as shown in Figure 4.1.

However, if we are to abandon the Boolean vision model, then we have to introduce some kind of continuous scale. We can clearly use a floating-point scale, for example, from 0 to 1, but then the scale must *mean* something. What does a value of 0.3 or 0.7 mean?

While it's tempting to think of the scale as representing probability, this is a flawed approach. Does 0.3 mean that the agent has a 30% chance of seeing an object? That's not really what humans experience and it has odd implications. How often should an agent roll a die to check if it sees an object? If the die is rolled often enough, eventually the object will be seen, even if it has a very low value, such as 0.1. Additionally, this isn't much different from the Boolean vision model, but it has the added detriment that it's more unpredictable and probably inexplicable to the player (the player will have difficulty building a model in their head of what the agent is experiencing; thus, the AI's behavior will appear random and arbitrary).

A better interpretation of the [0, 1] scale is for it to represent an *object identification certainty*. If the value is a little higher than zero, the agent sees the object but just isn't sure what it is. Essentially, the object is too blurry or vague to identify. As the value climbs toward one, then the certainty of knowing the identity of the object becomes much higher. This avoids any randomness and is more consistent with what humans experience. The numbers within Figure 4.1 show the object identification certainty within each vision zone, on a scale of 0–1.

### 4.2.1 Vision Sweet Spot

Human vision is most acute directly ahead and many games modify the traditional vision cone to reflect this. Further, it has been found to produce better results in games when this sweet spot first widens and then narrows with distance, as shown in Figure 4.1. Variations of this widening and then narrowing of ideal vision have been in use in games for a long time [Rabin 08] and have recently been used in games such as *Splinter Cell: Blacklist* [Walsh 15] and *The Last of Us* [McIntosh 15].

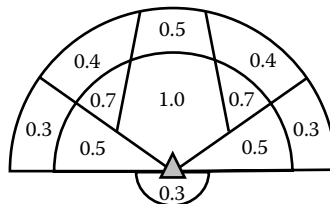


Figure 4.1

An example vision zone model with the associated object identification certainty values within each zone.

---

### 4.3 Incorporating Movement and Camouflage

If the zone values represent object identification certainty, then there are other aspects of an object that might affect this value other than location within a zone. For example, objects that are moving are much easier to spot than objects that are perfectly still. Additionally, objects are much harder to identify when they are camouflaged against their environment, because of either poor lighting or having similar colors or patterns to the background.

These additional factors can be incorporated into the vision model by giving an object a bonus or penalty based on the attribute. For example, if an object is moving, it might add 0.3 to its object identification certainty. If an object is camouflaged, perhaps it subtracts 0.3 from its object identification certainty. Therefore, if an object in the peripheral vision would normally have an object identification certainty of 0.3, then a camouflaged object in that same spot would essentially be invisible. Please note that the example zone value numbers and bonus/penalty values used here are for illustration purposes and finding appropriate numbers would depend on the particular game and would require tuning and testing.

### 4.4 Producing Appropriate Behavior from Identification Certainty

The final challenge is to figure out what the agent should do with these identification certainty values. With the simple vision model, it was simple. If the object was seen, it was 100% identified, and the agent would react. With this new vision model, it must be decided how an agent reacts when an object is only identified by 0.3 or 0.7, for example.

When dealing with partial identification, it now falls into the realm of producing an appropriate and nuanced reaction and selling it convincingly to the player. This will require several different animations and vocalizations. With low identification values, perhaps the agent leans in and squints toward the object or casually glances toward it. Note that as the agent adjusts their head, their vision model correspondingly turns as well. What once only had a 0.3 identification could shoot up to 0.8 or 1.0 simply by turning to look at it.

Another response to partial identification is to start a dialogue with the other agents, such as “Do you see anything up ahead?” or “I thought I saw something behind the boxes” [Orkin 15]. This in turn might encourage other agents to cautiously investigate the stimulus.

Deciding on an appropriate response to partial information will depend on both the game situation and your team’s creativity in coming up with appropriate responses. Unfortunately, more nuanced behavior usually requires extra animation and vocalization assets to be created.

#### 4.4.1 Alternative 1: Object Certainty with Time

An alternative approach is to require the player to be seen for some amount of time before being identified. This is the strategy used in *The Last of Us* where the player would only be seen if they were in the AI’s vision sweet spot for 1–2 s [McIntosh 15]. Although not implemented in *The Last of Us*, the object identification certainty values from Figure 4.1 could be used to help determine how many seconds are required to be seen.

---

#### 4.4.2 Alternative 2: Explicit Detection Feedback

A slightly different tactic in producing a more nuanced vision model is to explicitly show the player how close they are to being detected, as in the *Splinter Cell* series [Walsh 15]. *Splinter Cell* shows a stealth meter that rises as the player becomes more noticeable in the environment. This is a more gameplay-oriented approach that forces the player to directly focus on monitoring and managing the meter.

#### 4.5 Conclusion

Vision models that push for more realism allow for more interesting agent behavior as well as allow for more engaging gameplay. Consider how your game's vision models can be enhanced and what new kinds of gameplay opportunities this might provide. However, be ready to address with animations, vocalizations, or UI elements how this enhanced model will be adequately conveyed to the player.

#### References

- [McIntosh 15] McIntosh, T. 2015. Human enemy AI in *The Last of Us*. In *Game AI Pro<sup>2</sup>: Collected Wisdom of Game AI Professionals*, ed. S. Rabin. Boca Raton, FL: A K Peters/CRC Press.
- [Orkin 15] Orkin, J. 2015. Combat dialogue in *F.E.A.R.: The illusion of communication*. In *Game AI Pro<sup>2</sup>: Collected Wisdom of Game AI Professionals*, ed. S. Rabin. Boca Raton, FL: A K Peters/CRC Press.
- [Rabin 08] Rabin, S. and Delp, M. 2008. Designing a realistic and unified agent sensing model. In *Game Programming Gems 7*, ed. S. Rabin. Boston, MA: Charles River Media.
- [Walsh 15] Walsh, M. 2015. Modeling perception and awareness in *Splinter Cell: Blacklist*. In *Game AI Pro<sup>2</sup>: Collected Wisdom of Game AI Professionals*, ed. S. Rabin. Boca Raton, FL: A K Peters/CRC Press.