Virtual Agent Perception Combination in Multi Agent Based Systems

Dane M. Kuiper
University of Texas at Dallas
800 West Campbell Road
Richardson, Texas, USA
kuiper@utdallas.edu

Rym Z. Wenkstern
University of Texas at Dallas
800 West Campbell Road
Richardson, Texas, USA
rymw@utdallas.edu

ABSTRACT

In order to create realistic simulations,virtual agents need to learn about their environment through perception. To date, most multi-agent simulation systems that implement some form of perception have focused heavily on a single sense, vision. In this paper we discuss a multi-sense perception system for virtual agents situated in large scale open environments. The perception system consists of modules (i.e., sensors) for visual, audible and olfactory senses. It also includes a perception combination module that combines data received from the multiple sensors into useful knowledge.

Categories and Subject Descriptors

I.2.11 [**ARTIFICIAL INTELLIGENCE**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*

Keywords

Virtual Agent Perception; Multi Agent Simulation

1. INTRODUCTION

A perception system that accurately models the human sensory system is critical for simulating virtual agents evolving in open environments (i.e., inaccessible, non-deterministic, dynamic, continuous) [12]. Most Multi-Agent Based Simulation Systems (MABS) have tackled the challenge of perception by providing agents with global or complete local environmental knowledge. Even though this approach is straightforward and easy to implement, it is unfit to simulate realistic scenarios. Until recently, very few realistic perception techniques have been proposed. To date, most MABS that implement some form of perception have focused heavily on a single sense, vision. Since the integration of other senses such as smell or hearing is almost non-existent within MABS, the combination of perception data has drawn very limited attention.

In this paper, we present a perception combination algorithm. The algorithm has been fully implemented and is a core part of DIVAs, a multi agent systems framework. We discuss multiple case study scenarios and analyze the results of our perception combination. We also run experiments regarding combination execution time and evaluate these results.

In the remainder of this paper, we discuss related works in Section 2. In Section 3, we review background information on DIVAs. In Section 4, we introduce some fundamental concepts required for perception combination. In Section 5, we discuss the Combination Algorithm. In Section 6 we present our results. In Section 7, we provide some concluding comments and outline possible future work in our research area.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.*

2. RELATED WORK

2.1 Perception Combination in MABS

In most multi-agent simulation systems, virtual agents are provided with global knowledge of the environment [5, 2, 20, 1] and therefore do not require perception mechanisms. Of the systems that implement virtual agent perception, most regulate the agents by a single sense, vision. In [10], Pelechano et al. present a vision algorithm utilizing a "rectangle of influence." [13] details a vision algorithm involving agent attention and eye movement. In [11], a vision algorithm utilizing "view volumes" is explained. And in [18], Oijin and Dignum propose an interesting perception framework for BDI-agents using the vision sense. Unfortunately, no algorithms or results have been published yet.

In addition to vision, some attempts have been made to address auditory perception. In [11], Piza et al simulate agent hearing using 3D noise propagation waves, and in [3], Herrero and Antonio discuss an auditory perceptual model based on human auditory concepts such as the "focus of perception" and the "nimbus of projection." The model is detailed, but unfortunately, the implementation is rather underdeveloped and cannot be used to assess the accuracy and efficiency of the work. Finally, even though some work has been published on auditory perception, olfactory perception seems to have been neglected by researchers.

With respect to perception combination, only Piza et al. [11] discuss combination of sensory data, and the proposed solution consists of simply creating a union of the perceptions. In addition, no work on building a cohesive framework to integrate multiple senses together has been discussed in the literature.

In this paper we discuss a perception combination algorithm for virtual agents evolving in an open environment. Our agents are not provided with global knowledge but acquire environmental knowledge through their perception module which integrates various sensors. The sensor parameters

can be configured individually for each agent and can be modified at execution time. The perception combination algorithm combines vision, auditory and olfactory sensory data to produce knowledge.

2.2 Robotic Perception Combination:

There has been much work within the robotics community on perception and robotic perception combination [9, 19, 14, 4, 15]. The overall approach has similarities to our research. Within both robotic and virtual agent perception combination, combination of sense information begins with raw sensory data supplied by multiple sensors and utilizes perception combination algorithms in order to combine this raw data into useful knowledge. However, there are two key differences in the objectives of our research and robotic perception combination research: 1) robotic perception algorithms are provided with real perceived data by the sensors and the goal is to extract environmental information from this data. Our virtual agent perception algorithms are provided with environmental information by the sensors and the goal is to simulate perception. 2) Robotic perception and perception combination algorithms are designed using embedded processors with the goal to maximize accuracy for use by the robot. Our algorithms, however, are designed to maximize the efficiency of virtual software agents. Since our work involves the simulation of thousands of software agents in realtime, minimizing the execution time of perception and perception combination algorithms is our main goal. In robotic perception each processor has a one-to-one relationship with its robotic agent, whereas in our research, each processor has a one-to-thousands relationship with its agents.

2.3 Human Perception Combination:

Any discussion of perception or perception combination has background in biological beings. Schiffman [16] explains that in humans, sensation refers to "certain immediate and direct qualitative experiences - qualities or attributes such as "hard" "warm" "red" and so on - produced by simple isolated physical stimuli." For agents, this would represent the raw sensory data provided by each sensor. Schiffman explains perception as "the psychological process whereby meaning, past experiences, or memory and judgement are involved." This process closely resembles our perception module and perception combination techniques which combine the individual sensory data received by the sensors into meaningful knowledge using memorized event knowledge available in the agents knowledge base. The agent can make use of this new knowledge while planning.

The next section gives a brief overview of DIVAs.

3. DIVAS OVERVIEW

DIVAs (**D**ynamic **I**nformation **V**isualization of **A**gent **s**ystems) is a large scale distributed multi-agent system framework for the specification and execution of large scale distributed simulations where agents are situated in an open environment. DIVAs is based upon two underlying concepts.

- In order to manage a large distributed environment efficiently, it is necessary to partition the space into smaller defined areas called *cells*.
- Each cell is assigned a special agent called a *controller*. A cell controller is required to: 1) autonomously manage environmental information about its cell; and 2) provide local virtual agents with information about their surroundings.

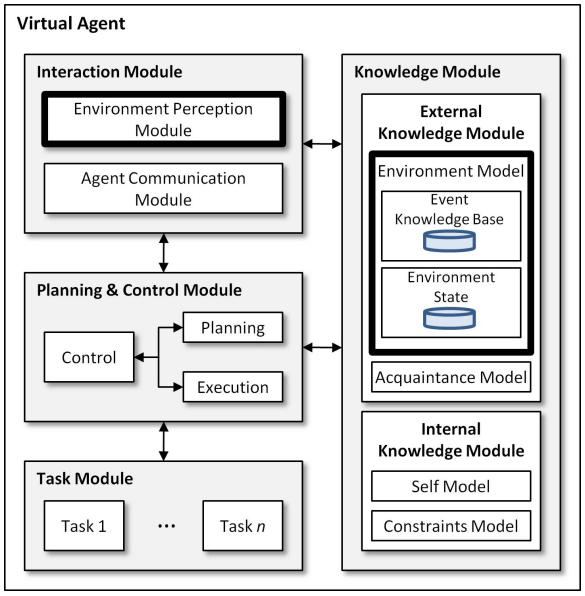

Figure 1: Conceptual View: DIVAs agent

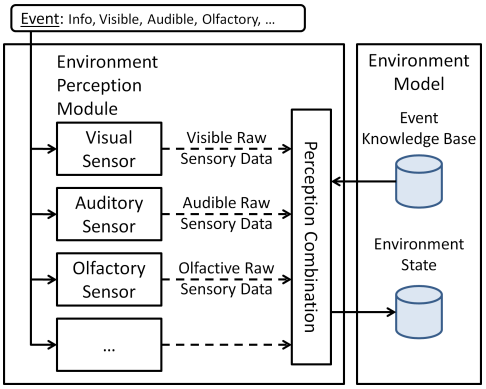

Figure 2: Perception Module with combination

In DIVAs, an agent consists of an *interaction module*, a *knowledge module*, a *task module*, and a *planning and control module* [8] (see Figure 1). The interaction module handles an agent's interaction with external entities and separates agent-environment interaction from agent-to-agent interaction. An agent communicates with other agents through the *agent communication module*. It receives environmental data from the cell controllers through the *environment perception module*. The environmental data includes agent states, object states as well as external event information. The processing of agent states, object states and simple events is addressed in [17]. In this paper we discuss external multi-sensory events and their perception by agents.

As shown in Figure 2, an agent perception module uses various perception sensors. The event data given to the agent includes information about the event as well as sensor interfaces. For example, an explosion includes audible, visible, olfactory, and tactile interfaces, whereas a siren only includes an audible interface. Once a sensor receives event

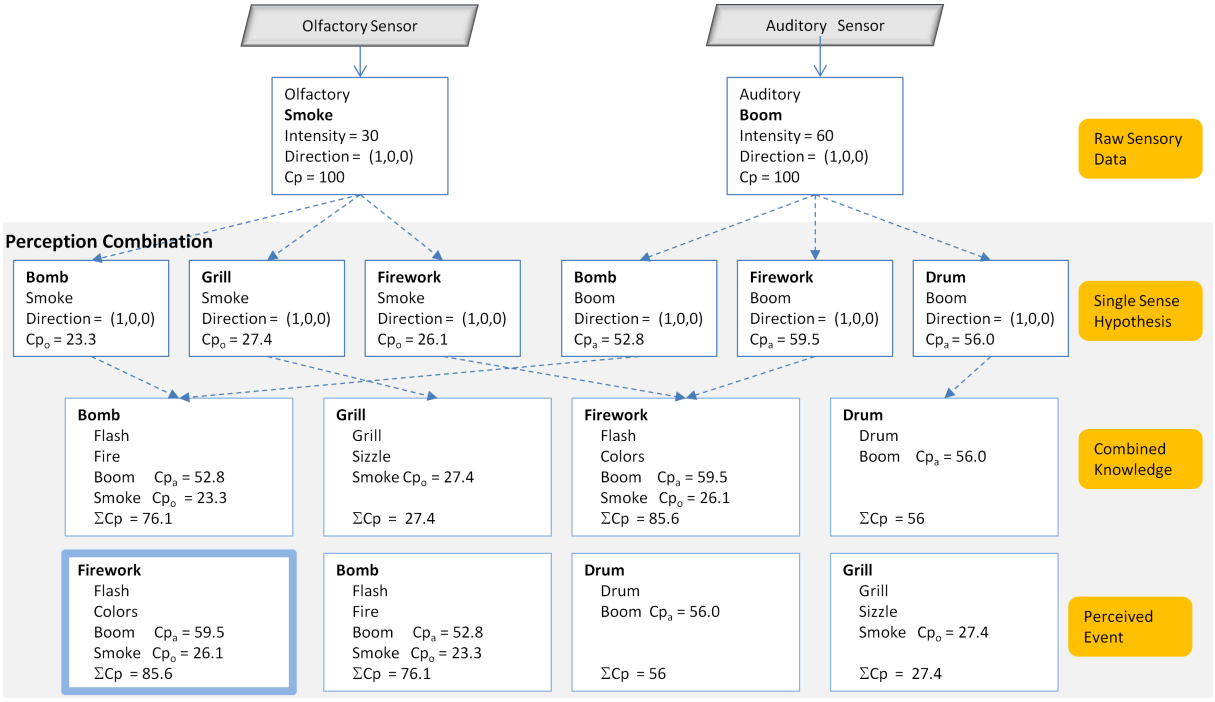

Figure 3: Perception Combination Data

data, it determines whether the information is perceivable by examining the interfaces. Using the agent current state, the sensor executes a specialized algorithm to determine what is perceived by the agent. For example, if a perception module receives an explosion message from the environment, it passes this information onto the various sensors. The vision sensor determines that the explosion implements the visible interface and therefore is a visible event. It extracts the event location and uses the agent's location, field of view and visible range to determine what is seen by the agent. The same applies to the other sensors. More information on the perception module and its use for single-sensory events is available in [17, 7].

The perception combination algorithm presented in this paper uses the raw sensory data produced by the vision, auditory and olfactory sensors. Before discussing the algorithm, we present some fundamental concepts.

4. SENSE COMBINATION CONCEPTS

4.1 Variable Definitions

Perception combination begins with the raw sensory data received from the various sensors. The data is combined and then converted into perceived events. Raw sensory data, combination data and perceived events all share some common variables:

1. The Name ($\mathcal{N}$) of the event. (e.g., bomb)
2. The Sense ($\mathcal{S}$) of the event. (e.g., vision)
3. The Type ($\mathcal{Y}$) of the event. (e.g., fire)
4. The Intensity ($\mathcal{I}$) of the event. This represents the power of the event when received by the sensor.
5. The Origin point ($\mathcal{O}$) of the event.
6. The Direction ($\mathcal{D}$) of the event. It is important to note that the vision sensor provides both $\mathcal{O}$ and $\mathcal{D}$ for an event, whereas auditory and olfactory sensors provide only $\mathcal{D}$.
7. The Minimum (m^-) expected $\mathcal{I}$ of the event/data.
8. The Maximum (m^+) expected $\mathcal{I}$ of the event/data.
9. The Certainty Percent ($\mathcal{C}_\mathcal{P}$) of the event. This represents the degree of certainty of the agent perception. (e.g., an agent's vision sensor reports seeing a fire: $\mathcal{S} = Vision$, $\mathcal{Y} = Fire$, $\mathcal{I} = 30$, $\mathcal{O} = (2, 2, 2)$ and $\mathcal{C}_\mathcal{P} = 100$).

Sample Event Knowledge

$\mathcal{N}$	$\mathcal{Y}$	$\mathcal{S}$	m^-	m^+
Bomb	Flash	Vision	40	200
	Fire	Vision	20	40
	Boom	Auditory	50	160
	Smoke	Olfactory	20	100
Firework	Flash	Vision	20	80
	Colors	Vision	40	100
	Boom	Auditory	40	120
	Smoke	Olfactory	5	40
Grilling Food	Grill	Vision	0	0
	Sizzle	Auditory	5	20
	Smoke	Olfactory	3	45
Drums	Drum	Vision	0	0
	Boom	Auditory	10	105
Spotlight	Flash	Vision	20	80

Figure 4: Sample event knowledge in agent KB.

4.2 Agent Event Knowledge Base

In order to successfully combine perception data, agents

need to make use of event knowledge stored in their Event Knowledge Base (EKB). EKB includes the following data $\mathcal{N}$, $\mathcal{Y}$, $\mathcal{S}$, m^- and m^+. Using the sample EKB given in Figure 4, an event will be perceived as a bomb with a high $\mathcal{C_P}$ if the agent sees a flash with intensity between 40 and 200, sees a fire with intensity between 20 and 40, hears a boom with intensity between 50 and 160 and smells smoke with intensity between 20 and 100. Each agent has its own EKB which is either populated by the simulation user or updated by the agent during the simulation. This knowledge may be incorrect, incomplete or non-existent. In addition, each agent assigns a Trust Constant (ξ) to each of its sensors. The trust constant allows the perception combination algorithm to place a higher value on data provided by sensors that the agent trusts more. Since each agent can have different knowledge about their environment, the results of perception combination can be completely different between agents.

5. PERCEPTION COMBINATION

5.1 Combination Algorithm Overview

Algorithm 5.1: PERCEPTION COMBINATION($L_1 : \rho List$)

$X \leftarrow$ EVENTENUMERATOR(L_1)
$(E_2, L_2) \leftarrow$ SINGLESENSEHYPOTHESISGENERATOR(L_1)
$L_3 \leftarrow$ HYPOTHESISCOMBINER(E_2, L_2)
$L_4 \leftarrow$ COMBINATIONSELECTOR(L_3, X)
$KnowledgeBase \leftarrow L_4$

The perception combination algorithm starts when *raw sensory data* is received from the sensors. Based on the sensory data *direction* and *origin*, the algorithm first determines whether the data is related to one or more events. This is achieved by executing the *EventEnumerator*. In the example given in Figure 3, *smoke* and *boom* have the same direction. Therefore, the *EventEnumerator* deduces that it is very likely that the two perceptions are related to the same event.

Following this step, the agent's Event Knowledge Base (EKB) is queried to determine the likelihood that a single raw sensory data is related to a known event. For example, in Figure 3, (olfactory,*smoke*, $\mathcal{I} = 30$), is matched against the agent EKB given in Figure 4. This identifies the known events that could be related to the raw sensory data and creates *single sense hypotheses*: a *bomb* with $\mathcal{C_P}$ of 23.3, a *grill* with $\mathcal{C_P}$ of 27.4 or a *firework* with $\mathcal{C_P}$ of 26.1. This step is implemented by the *SingleSenseHypothesisGenerator* and is executed on all raw sensory data.

In the next step, the single sense hypotheses related to the same event are grouped, and the total $\mathcal{C_P}$ for that event is computed. This is referred to as *combined knowledge*. For example, in Figure 3, hypotheses (bomb,smoke, $\mathcal{C_P} = 23.3$) and (bomb,boom, $\mathcal{C_P} = 52.8$) related to *bomb* are grouped, and the total $\mathcal{C_P}$ for bomb ($\mathcal{C_P}$=76.1) is computed. The list of combined knowledge for all possible events is sorted in decreasing order using the total $\mathcal{C_P}$. This step is executed by the *HypothesisCombiner*.

Finally, based on the *EventEnumerator*'s output value, the top events from the ordered list are selected and become the agent's *perceived events*. In Figure 3, since the *EventEnumerator* determined that the perceived raw sensory data are likely to be related to one event, the first event in the list is selected. In this case the agent's perceived event is a *firework*. This information is stored in the agent's Environment State Knowledge Base and used for planning and decision making. The perception combination algorithm is shown in Algorithm 5.1.

Algorithm 5.2: STEP 1 PROCEDURES

```
procedure EVENT ENUMERATOR(L1 : ρList)
 if L1.size > 0
   then diff.add(L1[0])
 for i = 0 to L1.size
 {  new ← true
    for j = 0 to diff.size
    {  if SOURCECOMPARISON(L1[i], diff[j]))
       {   then new ← false
    if new = true
      then diff.add(L1[i])
 r ← diff.size
 return (r)

procedure SOURCECOMPARISON(o1 : ρ, o2 : ρ)
 if HASORIGIN(o1) and HASORIGIN(o2)
   then { if o1.origin ≈ o2.origin
          then return ( true )
          else return ( false )
 else if HASDIRECTION(o1) and HASDIRECTION(o2)
   then { if o1.direction ≈ o2.direction
          then return ( true )
          else return ( false )
 else if HASDIRECTION(o1) and HASORIGIN(o2)
   then { if o1.direction ≈ GETDIRECTION(o2.origin)
          then return ( true )
          else return ( false )
 else if HASORIGIN(o1) and HASDIRECTION(o2)
   then { if GETDIRECTION(o1.origin) ≈ o2.direction
          then return ( true )
          else return ( false )
 else return ( false )
```

5.2 Perception Data

As mentioned earlier, the perception combination algorithm uses four types of data: Raw Sensory Data (ρ), Single Sense Hypothesizes (η), Combined Knowledge (σ) and Perceived Events ($\mathcal{F}$) (see Figure 3).

- **Raw Sensory Data (ρ):** This is the perception data provided directly by each sensor. Since this is concrete data, we assign $\mathcal{C_P}$ to 100. Each raw sensory data ρ contains: $\mathcal{S}$, $\mathcal{Y}$, $\mathcal{I}$, $\mathcal{C_P}$ and either $\mathcal{O}$ or $\mathcal{D}$ depending on the sensor that sensed it.
- **Single Sense Hypothesis (η):** This is a hypothesis that a particular known event may be related to a raw sensory data. Each η contains: $\mathcal{N}$, $\mathcal{Y}$, $\mathcal{C_P}$ and either $\mathcal{O}$ or $\mathcal{D}$.
- **Combined Knowledge (σ):** This is the knowledge resulting from the grouping of Single Sense Hypothesizes. Each σ contains: $\mathcal{N}$, $\mathcal{C_P}$ and either $\mathcal{O}$ or $\mathcal{D}$.

- **Perceived Event ($\mathcal{F}$):** This corresponds to the event that an agent perceived as a result of the perception combination algorithm. Each $\mathcal{F}$ contains: $\mathcal{N}$, $\mathcal{C_P}$ and either $\mathcal{O}$ or $\mathcal{D}$.

Algorithm 5.3: STEP 2 PROCEDURES

```
procedure SINGLESENSEHYPOTHESISGENERATOR(
L1 : ρList)
 for each ρ ∈ L1
 do { EK ← EKB.getEventKnowledgeByType(ρ)
      for each e ∈ EK
      do { if e.S = ρ.S and if e.Y = ρ.Y
           then { α ← (m⁻ + m⁺)/2
                  ℜ ← |m⁻ − m⁺|
                  ε ← ℜ/2
                  v ← ρ.I
                  if |α − v| ≥ ℜ
                    then {C_P ← 0
                  if |α − v| ≥ ε
                    then {C_P ← 70 − (|α − v| − ε)/ε · 70
                  if |α − v| < ε
                    then {C_P ← 100 − |α − v|/ε · 30
           if C_P > 0
           then { C_P ← C_P · ρ.ξ
                  η.C_P ← C_P
                  η.N ← e.N
                  η.Y ← e.Y
                  η.O ← ρ.O
                  η.D ← ρ.D
                  r ← KB.getEvent(η.N)
                  r.O ← η.O
                  r.D ← η.D
                  if r ∉ E2
                    then {E2.add(r)
                  L2.add(η) } } }
 return(E2, L2)
```

In the next section, we discuss the perception combination algorithm in further detail.

5.3 Perception Combination in Detail

Step 1 - The Event Enumerator takes a list of raw sensory data (ρ) and returns an integer, the estimated number of events that occurred. It uses the ρ's origin or direction to determine the proximity of the sensed data. Events that are in "close proximity" (e.g., less than three units apart) are considered to come from the same event. The Event Enumerator is shown in Algorithm 5.2.

Step 2 - The Single Sense Hypothesis Generator takes a list of ρ and returns a list of single sense hypotheses (η). It uses the agent's Event Knowledge Base (EKB) to individually associate each ρ with the known event (e) most likely to have caused that sensation. In order to create such hypothesizes, the Single Sense Hypothesis Generator matches the raw sensory data's $\rho.\mathcal{Y}$ and $\rho.\mathcal{S}$ with known events' $e.\mathcal{Y}$ and $e.\mathcal{S}$. If there is a match between ($\rho.\mathcal{Y}$,$\rho.\mathcal{S}$) and ($e.\mathcal{Y}$,$e.\mathcal{S}$), it calculates the $\mathcal{C_P}$ using a six segment linear distribution defined by the average α, range $\Re$ and extents ε. If $\mathcal{C_P}$ is not zero, a new η is created and added to L_2. Next, an empty record (r) of the event associated with η is added to E_2. After all raw data is processed, L_2 and E_2 are returned. The algorithm for Single Sense Hypothesis Generator is shown in Algorithm 5.3. The Single Sense Hypothesis Generator worst case runtime is $O(M \cdot n)$ time, where M is the number of events in the EKB and n is the number of ρ in L_1.

Algorithm 5.4: STEP 3 PROCEDURES

```
procedure HYPOTHESISCOMBINER(E2 : rList, L2 : ηList)
 for each r ∈ E2
 do { C_P ← 0
      i ← 0
      EK ← r.getEventKnowledgeByName()
      maxEvents ← EK.size()
      for each ek ∈ EK
      do { for each η ∈ L2
           do { if r.N = η.N and if ek.Y = η.Y
                and if SOURCECOMPARISON(r, η)
                then { C_P ← C_P + η.C_P
                       i ← i + 1
                       BREAK } } }
      if i < maxEvents
        then {C_P ← C_P · i/maxEvents
      σ.N ← r.N
      σ.O ← r.O
      σ.D ← r.D
      σ.C_P ← C_P
      L3.add(σ) }
 return(L3)
```

Step 3 - The Hypothesis Combiner takes lists L_2 and E_2 produced by the Single Sense Hypothesis Generator and returns a list L_3 of Combined Knowledge (σ). For each r in E_2, it finds all the hypotheses η that have the same name ($\mathcal{N}$) and type ($\mathcal{Y}$) as r's. If these η are within a "close proximity" of each other, they are combined to create a new σ. This new σ is then added to L_3. L_3 is sorted by $\mathcal{C_P}$ in decreasing order. The algorithm for Hypothesis Combiner is shown in Algorithm 5.4. Since event knowledge is constant in size, the Hypothesis Combiner worst case runtime is $O(E \cdot n)$ time, where E is the number of events in E_2 and n is the number of η in L_2.

Algorithm 5.5: STEP 4 PROCEDURES

```
procedure COMBINATIONSELECTOR(L3 : σList, X : int)
 for 1 to X
 do { σ ← L3.getGreatest()
      F.N ← σ.N
      F.O ← σ.O
      F.D ← σ.D
      F.C_P ← σ.C_P
      L4.add(F) }
 return(L4)
```

Step 4 - The Combination Selector takes the list L_3 produced by the Hypothesis Combiner and X produced by the

Example Event Properties			
$\mathcal{N}$	$\mathcal{Y}$	$\mathcal{I}$	$\mathcal{S}$
Bomb	Flash	90	Vision
	Fire	30	Vision
	Boom	95	Auditory
	Smoke	50	Olfactory
Firework	Flash	40	Vision
	Colors	50	Vision
	Boom	50	Auditory
	Smoke	20	Olfactory
Drums	Drums	-	Vision
	Boom	25	Auditory
Grilling Food	Grill	-	Vision
	Sizzle	12	Auditory
	Smoke	8	Olfactory
SpotLight	Flash	50	Vision

Figure 5: Sample event properties in simulation.

Event Enumerator and returns a list L_4 of perceived events ($\mathcal{F}$). It extracts the X combined knowledge records from L_3 with the highest $\mathcal{C}_\mathcal{P}$ and creates X new $\mathcal{F}$. The list L_4 of these perceived events is then passed onto the agent's Environment State Knowledge Base for use in planning and decision making. The algorithm for Combination Selector is given in Algorithm 5.5.

This concludes the discussion of the DIVAs perception combination algorithm. The next section discusses our experimental results.

6. RESULTS

In this section we first discuss the execution of our combination algorithm on various test case scenarios, then we present algorithm execution time results.

6.1 Test Case Scenario Results

For testing combination results, we have implemented five basic environment events: bombs, fireworks, grills, drums and spotlights. For each event, we have assigned the properties given in Figure 5. We notice that the events have very similar raw sensory data. For example, bombs, fireworks and drums all generate loud boom sounds.

We populate the agent's event knowledge base with the information shown earlier in Figure 4. The agent's trust constant values are set for each sensor as follows: Vision = .95, Auditory = .7 and Smell = .3. We chose these values since people usually tend to rely mostly on their vision.

In the remainder of this section, we present four scenarios. More test case scenarios and results are available in [6]. For each scenario, we first describe the event; then we explain how we expect the algorithm to perform; next we execute the scenario in DIVAs and present a simulated virtual agent's output. More precisely, we show the output produced by the Hypothesis Combiner and the Combination Selector. Finally we give a brief analysis of the results.

Scenario 1

Event: A spotlight, drums and grilling food events are triggered at different locations near an agent.

Expected Agent Reaction: The agent will perceive the following raw sensory data: a flash, a boom, sizzle and smoke. The three source locations are different. The perception combination system should able to correctly determine that there are three distinct events and assign them appropriately.

Actual Agent Reaction: The Event Enumerator detected three events. The full results from the Hypothesis Combiner and the Combination Selector are given in Figure 6. The agent correctly sensed all three events.

Analysis: Perception combination can be used to successfully determine what events occurred in scenarios with multiple events. One can also notice that the perception combination module considered the possibility that all three events could be bombs.

Scenario 1		
	$\mathcal{N}$	$\mathcal{C}_\mathcal{P}$
Step 3	Grilling Food	186.7428
	Drums	150.6315
	Spotlight	95.0000
	Bomb	17.5156
	Firework	16.6250
	Firework	7.6562
	Bomb	6.6819
	Firework	5.6357
	Bomb	3.6750
Step 4	Grilling Food	186.7428
	Drums	150.6315
	Spotlight	95.0000

Figure 6: Step 3-4 Results for Scenario 1

Scenario 2		
	$\mathcal{N}$	$\mathcal{C}_\mathcal{P}$
Step 3	Grilling Food	186.7428
	Drums	150.6315
	Spotlight	95.0000
	Firework	89.7508
	Bomb	83.6173
Step 4	Grilling Food	186.7428

Figure 7: Step 3-4 Results for Scenario 2

Scenario 2

Event: A spotlight, drums and grilling food events are triggered at the same location in front of the agent.

Expected Agent Reaction: The agent perceives raw sensory data similar to Scenario 1. Due to the close event proximity, the perception combination system should not be able to correctly determine that there were three distinct events. It will only detect one event.

Actual Agent Reaction: The Event Enumerator detected one event. The full results from the Hypothesis Combiner and the Combination Selector are given in Figure 7. Grilling food was the most likely result.

Analysis: When many events occur at the same location, the perception combination module may not be able to correctly determine the exact number of events that occurred. In such cases the perception combination algorithm selects the most likely events. It is also important to notice that in comparison with Scenario 1, the likelihood of a bomb or

Scenario 3		
	$\mathcal{N}$	$\mathcal{C}_\mathcal{P}$
Step 3	Bomb	184.1207
	Spotlight	44.3333
	Drums	26.7105
	Firework	15.5312
Step 4	Bomb	184.1207

Figure 8: Step 3-4 Results for Scenario 3

Scenario 4		
	$\mathcal{N}$	$\mathcal{C}_\mathcal{P}$
Step 3	Bomb	15.1136
	Drums	11.6052
	Firework	9.1875
Step 4	Bomb	15.1136

Figure 9: Step 3-4 Results for Scenario 4

firework exploding is far higher due to the close proximity of the raw sensory data.

Scenario 3

Event: In an attempt to try to trick an agent,we trigger an explosion, except the bomb produces no smoke.

Expected Agent Reaction: The agent will perceive a flash, fire and a boom, but no smoke. The final result should not change with just the lack of smoke.

Actual Agent Reaction: The Event Enumerator detected one event. The full results from the Hypothesis Combiner and the Combination Selector are given in Figure 8. The agent detected a bomb.

Analysis: For comparison purposes, the $\mathcal{C}_\mathcal{P}$ of the final combination of a normal bomb is 273.24. The lack of smoke lowered $\mathcal{C}_\mathcal{P}$ a bomb exploded, but the perception combination module was still highly certain that an explosion occurred based on the rest of the raw sensory data.

Scenario 4

Event: A large bookcase crashes over behind an agent. This event is not in the agent's event knowledge base. Assume a bookcase crash event has a boom sound with intensity 100.

Expected Agent Reaction: The agent will perceive a very loud boom. It should process all events that could generate a loud boom and choose the most likely.

Actual Agent Reaction: The Event Enumerator detected one event. The full results from Step 3-4 are in Figure 9. The agent decided the most likely event was a bomb, but $\mathcal{C}_\mathcal{P}$ was very low.

Analysis: Due to the lack of event knowledge a bookcase crashing over, the agent was unable to determine what occurred. It incorrectly predicted that a bomb exploded, but assigned a low $\mathcal{C}_\mathcal{P}$, since the hypothesis that a bomb exploded was based only on the auditory sense.

6.2 Algorithm Execution Time Results

All of the following tests were executed on a single system using only a single core of an Intel i7 X980 CPU (3.33GHz).

6.2.1 Scalability in Number of Agents

We tested the effects of agent perception combination on the execution time of a simulation with a varying number of agents. For this test set, each agent receives exactly four events. The no-combination case receives four events that are simply passed onto the Environment State Knowledge Base, whereas the combination case must combine the four events. Full results are shown in Figure 10. The result-time is in milliseconds. The results clearly show that for a small number of events (four in this case), combination adds almost no time to non-combination agent perception calculations. In addition, as the number of agents combining events was increased, the time required for all agents to perceive and combine perception data was sub-linear. A single system could comfortably simulate the perception and combination of well over 1000 agents in simulated realtime. Using the DIVAs simulation system's decentralized structure, these results support that perception combination is feasible in very large scale simulations involving 10,000s or even 100,000s of agents.

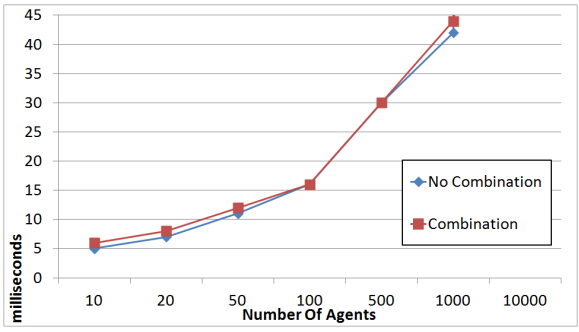

Figure 10: Scalability Results Graph for four events.

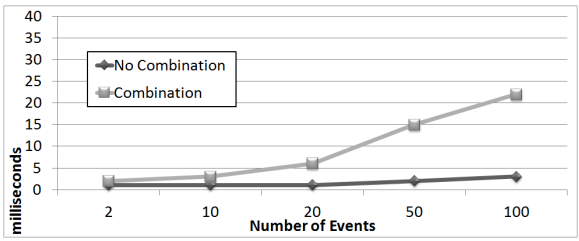

Figure 11: Scalability Results Graph for one agent.

6.2.2 Scalability in Number of Events - One Agent

We tested the effects of agent perception combination on the time it takes for a single agent to execute. For this set of tests, the number of agents (one) is held constant. Full results are shown in Figure 11. The result-time is in milliseconds. Up to 100 simultaneous events combined, the algorithm performs extremely well. These results, are very promising, since in our testing there are frequently less than 30 simultaneous events occurring in the same perceivable area at the same time.

6.2.3 Scalability in Number of Events - 100 Agents

We tested the effects of agent perception combination on the time it takes 100 agents to execute. Full results are shown in Figure 12. The result-time is in milliseconds. One agent requires 6ms in order to do perception combination on 20 events. 100 agents only requires 42ms in order to do

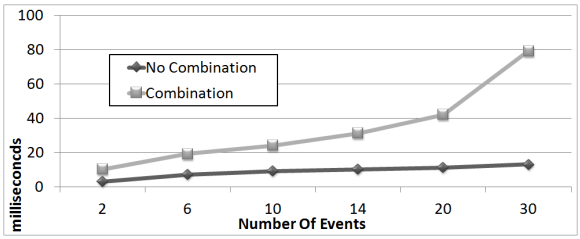

Figure 12: Scalability Results Graph for 100 agents.

perception combination on 20 events. Similar to the above tests, these results are very promising for large scale multi-agent based simulations.

7. CONCLUSION

In this paper we discussed Virtual Agent perception combination in large scale multi-agent based simulation systems. We provided results displaying the performance of our perception combination system in multiple scenarios and also showed that the algorithm maintains good execution time even with large numbers of agents. While we only presented results based on the vision, hearing and smell senses, our perception combination algorithm can make use of any number of senses. Future work includes implementing further optimizations of our perception combination system. We would also like to implement features that would allow agents to rely more on highly trusted senses.

Combination of virtual agent perceptions remains a new area of research, with great potential for new ideas. We plan to continue to validate our combination algorithm on new scenarios and improve upon our results.

8. REFERENCES

[1] S. Bandini, M. Federici, S. Manzoni, and G. Vizzari. Pedestrian and crowd dynamics simulation: Testing sca on paradigmatic cases of emerging coordination in negative interaction conditions. *Parallel Computing Technologies*, pages 360–369, 2007.

[2] B. Banerjee, A. Abukmail, and L. Kraemer. Advancing the layered approach to agent-based crowd simulation. In *Proceedings of IEEE Workshop on Parallel and Distributed Simulation*, pages 185–192, Rome, Italy, June 3-6 2008.

[3] P. Herrerro and A. de Antonio. Introducing human-like hearing perception in intelligent virtual agents. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS03)*, pages 7–14, Melbourne, Australia, July 14 - 18 2003.

[4] K. Itoh, H. Miwa, Y. Nukariya, H. Takanobu, and A. Takanishi. A new memory model for humanoid robots-introduction of associative memory using chaotic neural network. In *Robot and Human Interactive Communication, 2004.*, pages 125–130. IEEE, 2004.

[5] J.Shi, A. Ren, and C.Chen. Agent-based evacuation model of large public buildings under fire conditions. *Automation in Construction*, 18(3):338–347, May 2009.

[6] D. Kuiper. *Agent Perception in Multi-Agent Based Simulation Systems*. PhD thesis, University of Texas at Dallas, November 2012.

[7] D. Kuiper and R. Wenkstern. Virtual agent perception in large scale multi-agent based simulation systems. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1235–1236. AAMAS, 2011.

[8] R. Z. Mili, R. Steiner, and E. Oladimeji. DIVAs: Illustrating an abstract architecture for agent-environment simulation systems. *Multiagent and Grid Systems, Special Issue on Agent-oriented Software Development Methodologies*, 2(4):505–525, January 2006.

[9] M. Munz, M. Maĺ andhlisch, and K. Dietmayer. Generic centralized multi sensor data fusion based on probabilistic sensor and environment models for driver assistance systems. *Intelligent Transportation Systems Magazine, IEEE*, 2(1):6 –17, spring 2010.

[10] N. Pelechano, J. Allbeck, and N. Badler. Controlling individual agents in high-density crowd simulation. In *2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 99–108, San Diego, California, August 02 - 04 2007.

[11] H. Piza, F. Ramos, and F. Zuniga. Virtual sensors for dynamic virtual environments. In *Computational Advances in Multi-Sensor Adaptive Processing, 2005 1st IEEE Int. Workshop*, pages 177–180. IEEE, 2005.

[12] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[13] S. J. Rymill and N. A. Dodgson. Psychologically-based vision and attention for the simulation of human behaviour. In *Proceedings of Computer graphics and interactive techniques*, pages 229–236, Dunedin, New Zealand, November 29 - December 02 2005.

[14] B. Scassellati. *High level perceptual contours from a variety of low level features*. PhD thesis, MIT Department of Electrical Engineering and Computer Science, 1995.

[15] B. Scassellati. *Foundations for a Theory of Mind for a Humanoid Robot*. PhD thesis, Massachusetts Institute of Technology, 2001.

[16] H. Schiffman. *Sensation and perception*. Wiley, 1976.

[17] T. Steel, D. Kuiper, and R. Wenkstern. Virtual agent perception in multi-agent based simulation systems. In *Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT-10)*, pages 453–456, Toronto, Canada, August 2010. ACM.

[18] J. van Oijen and F. Dignum. A perception framework for intelligent characters in serious games. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '11, pages 1249–1250, Richland, SC, 2011.

[19] Z. Xiang. An environmental perception system to autonomous off-road navigation by using multi-sensor data fusion. In *Neural Networks and Brain, 2005. ICNN B '05. International Conference on*, volume 2, pages 1221 –1226, oct. 2005.

[20] M. Xiaofeng, W. Chaozhong, and Y. Xinping. A multi-agent model for evacuation system under large-scale events. In *Proceedings of International Symposium on Computational Intelligence and Design(ISCID08)*, pages 557–560, Wuhan, China, October 17-18 2008.