SPREAD : Sound Propagation and Perception for Autonomous Agents in Dynamic Environments

Pengfei Huang
University of Pennsylvania

Mubbasir Kapadia
University of Pennsylvania

Norman I. Badler*
University of Pennsylvania

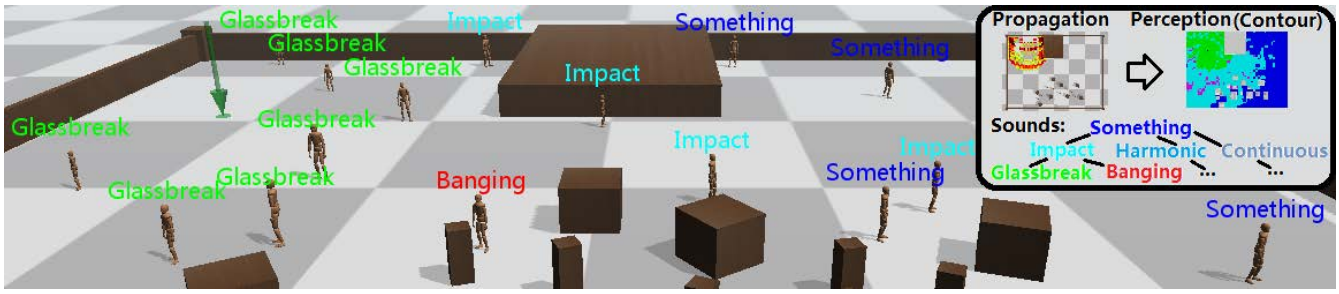


Figure 1: *Agent-based sound perception using packet representation and propagation model. The green arrow in the scene is the sound source position, and the agents' captions on top show what they just heard. Green indicates correct perception, blue and cyan indicate approximate perception, and red is an incorrectly perceived signal. These sound candidates or categories are from a sound cluster structure.*

Abstract

The perception of sensory information and its impact on behavior is a fundamental component of being human. While visual perception is considered for navigation, collision, and behavior selection, the acoustic domain is relatively unexplored. Recent work in acoustics focuses on synthesizing sound in 3D environments; however, the perception of acoustic signals by a virtual agent is a useful and realistic adjunct to any behavior selection mechanism. In this paper, we present SPREAD, a novel agent-based sound perception model using a discretized sound packet representation with acoustic features including amplitude, frequency range, and duration. SPREAD simulates how sound packets are propagated, attenuated, and degraded as they traverse the virtual environment. Agents perceive and classify the sounds based on the locally-received packet set using a hierarchical clustering scheme, and have individualized hearing and understanding of their surroundings. Using this model, we demonstrate several simulations that greatly enrich controls and outcomes.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: Autonomous Agent, Sound Representation, Sound Propagation, Sound Perception

1 Introduction

Autonomous agent animation research models vision-based perception of agents using abstract perception queries such as line-of-sight ray casts and view cone intersections with the environment. However, if we want our virtual agents to behave even more human-like they ought to have hearing models to perceive and understand the acoustic world. Moreover, sound propagates differently from light, providing a rich set of additional perceptual options for an agent, including perception and possible localization of an unseen event, and the recognition or possible mis-identification of the sound type. For example, a person may not be seen because of visual occlusion, but the person's footsteps or voice may still be heard. In a cocktail party, someone might not be able to see everyone else, but hearing her name uttered might immediately command her attention.

For virtual reality and games with autonomous agents, acoustic perception can provide useful behaviors including possible goals (sound sources), avoidance regions (noisy areas), knowledge of unseen events (shots), or even navigation cues (such as hearing someone approaching around a blind corner). Virtual agents with 'ears' can greatly improve the realism of crowd models, games and virtual reality systems.

Prior work [Takala and Hahn 1992; Savioja et al. 1999; James et al. 2006] has developed computational models for sound synthesis and propagation, with limited work that factors the perception of sound into agent behavior. Visual perception is used for agent steering, e.g., [Ondřej et al. 2010], while audio perception is uncommon [Monzani and Thalmann 2000].

Our approach to sound modeling, propagation and perception as illustrated in Figure 2. First, we describe a minimal yet sufficient set of acoustic features to characterize the human-salient components of a sound signal [Gygi et al. 2007]. These features include amplitude, frequency, and duration, which are found to be strongly correlated to sound classification. Second, we develop a real-time sound packet propagation and distortion model using adaptive 2D quad-tree meshes with pre-computed propagation values suitable for dynamic virtual environments.

During an offline process, we build a sound database using a discrete sound packet representation, and group similar sounds using Hierarchical Clustering Analysis (HCA) for agent sound perception. During simulation, sound packets are propagated through the scene based on the Transmission Line Matrix method [Kagawa et al. 1998; Kristiansen and Viggen 2010], which accounts for sound packet degradation based on distance traveled, and absorption and reflection by obstacles and moving agents in the envi-

*e-mail: {pengfei, mubbasir, badler}@seas.upenn.edu

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SCA 2013, July 19 – 21, 2013, Anaheim, California.
Copyright © ACM 978-1-4503-2132-7/13/07 $15.00

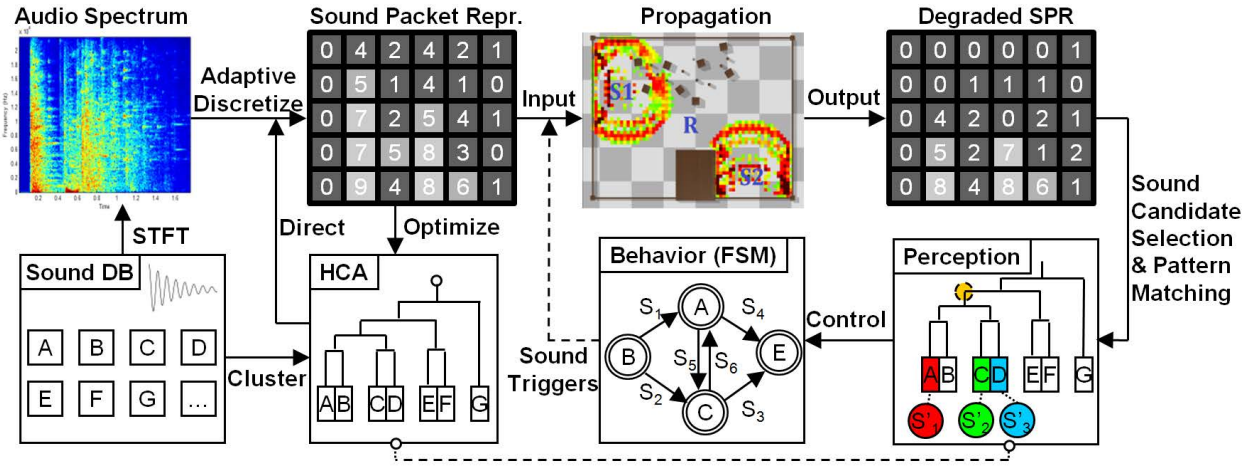


Figure 2: *SPREAD Framework Overview. (a) Sound signals in a database are adaptively discretized in the frequency and time domain to precompute a minimal and sufficient sound packet representation for each sound, and hierarchically clustered based on a human perceived similarity measure. (b) Sounds are propagated in dynamic environments using the Transmission Line Matrix Method to simulate natural sound degradation. (c) Degraded sound packets are perceived using hierarchical clustering analysis to model approximate human-like perception using sound categories. (d) Auditory triggers are incorporated into agent architectures to enhance the behavioral realism of autonomous virtual humans.*

ronment. To reduce computational costs, we add a quad-tree-based pre-computation to accelerate the propagation model. The original algorithm can easily be ported to the GPU for further acceleration. Agents receive a series of altered sound packets and use Dynamic Time Warping algorithms to identify similar sounds from the HCA. If multiple sounds from the HCA are above a similarity threshold, their Lowest Common Ancestor (i.e., the more general sound category) is perceived. Using this framework, virtual agents possess individual hearing. Our main contributions are:

- An adaptive discretization of continuous sound signals to obtain a minimal, yet sufficient sound packet representation (SPR) necessary for human-like perception, and a hierarchical clustering scheme to facilitate approximate perception.
- Efficient planar sound propagation of discretized sound signals which exhibits acoustic properties such as attenuation, reflection, refraction, and diffraction, as well as multiple convoluted sound signals.
- Agent-based sound perceptions using hierarchical clustering analysis that accommodates natural sound degradation due to audio distortion and facilitates approximate human-like perception.

Experimental results show that our propagation framework works efficiently for multiple and different sound signals in dynamic virtual environments. A sound signal is easily identified if the agent is close to the source or a sound is less attenuated, absorbed, or reflected in the scene; conversely a sound is difficult to identify as sound packets suffer from interval degradation and overlapping effects. Our sound propagation methodology is not just based on distance, but takes into account the static environment, dynamic (e.g., other agents) features, and packet content degradation. We integrate SPREAD into agent attention and behavior models and demonstrate several novel game-like simulations that greatly enhance both play and user experiences. Our method is not intended for auralization, but serves as a companion to auralization – enabling virtual agents to hear and classify sounds much like their real human counterparts. Auralization is not a pre-requisite for this capability.

2 Related Work

Sound in Autonomous Agent Modeling and Game Applications. Virtual human research [Pelechano et al. 2008; Thalmann 2007] aims to simulate interacting autonomous agents. Commercial tools [Unity3D 2012] provide intuitive user interfaces to modify different sound features for sound synthesis, however, no tools exist to recreate more human-like sound perception.

Sound Synthesis and Features. Since the seminal work of [Takala and Hahn 1992], sound synthesis models [James et al. 2006] have been proposed for complex physical phenomena such as rigid body fracture and object contacts. [Xu et al. 2005] details many important audio feature computations: e.g., root mean square, frequency, centroid, and duration. Here we use amplitude, frequency range, and duration as a simplified but perceptually adequate basis for an environmental sound packet representation.

Sound Propagation and Degradation. Sound is propagated in virtual environments using beam tracing [Funkhouser et al. 1998] or frustum tracing [Chandak et al. 2008]; these methods treat the sound signal as rays and are unable to model acoustic properties such as diffraction. The work in [Raghuvanshi et al. 2009] uses an adaptive rectangular decomposition of a 3D scene and exploits graphics hardware to efficiently simulate sound propagation in complex virtual environments. The Transmission Line Matrix (TLM) method [Kristiansen and Viggen 2010] uses cellular automata to model sound propagation in a uniformly discretized environment and can demonstrate effects such as diffraction, absorption and reflection. In [Savioja et al. 1999], both numerical and geometric methods are used to construct a virtual acoustics environment with auralization and signals degradation. The empirical sound absorption rate for many materials is documented in [Mast 2000]. Sound intensity attenuation is often approximated as a quadratic function of distance [Holland et al. 1998].

Sound Perception and Behavior Models. Human factors experiments [Gygi et al. 2007] have been conducted to understand the relevance of sound properties for sound similarity: they conclude that amplitude, duration, and frequency strongly correlate with the principal components of sound classification. Based on these hu-

man judgements a hierarchical organization of 100 environmental sound signals is generated which clusters sounds that were perceived to be similar. Other work has considered human voice signals: e.g., [Monzani and Thalmann 2000] proposes an approximate sound propagation model to simulate how agents communicate via speech signals which experience only amplitude reduction and signal-noise-ratio effects. Herrero et al. [Herrero and de Antonio 2003] model sound perception in virtual agents by considering sound localization, the sound pressure level of the human voice, and the clarity of the perceived signal. However, the effects of propagation are not modeled and the understanding of speech signals is based on fixed thresholds. For our work we focus on the Gygi et al. environmental sound set as its perceptual classification is available; this removes consideration of language, speech qualities, content (semantics) and comprehension. For simulation, perceptions should map to behaviors, and a perceptual model could include various sensing modalities [Cony et al. 2007].

3 Sound Categorization and Representation

Sounds are continuous signals that are typically represented as 1D wave forms. A discretized sound representation must sufficiently capture the distinguishing properties of different signals and facilitate efficient sound propagation in complex environments while exhibiting appropriate sound degradation. This sound data representation will be received by agents who apply human-like sound perception models that determine whether any identifiable sound or sounds have been heard and, if so, what sound type or category they appear to represent. In signal processing, a large number of features are used to represent sound for signal analysis [Xu et al. 2005]. Human perception, however, is usually correlated to a small subset of features for environmental sounds [Gygi et al. 2007], such as frequency, amplitude, and duration.

3.1 Sound Feature Selection and Categorization

Sounds attenuate and degrade due to the environmental influences of reverberation, reflection, and diffraction. These effects cause sound signals to degrade in a non-linear fashion, resulting in complete attenuation, lack of perceptual specificity, or even incorrect classification of sound signals. For example, a ship noise may be perceived as a generic mechanical noise, possibly mis-identified as a construction noise which is perceptually similar, but would never be misinterpreted as a harmonic sound such as a siren.

Gygi et al. [Gygi et al. 2007] investigated human categorization of 100 sounds with an average of 1 second duration, providing a representative sound database of common environmental sounds. The subjects were required to rate the similarity between any two of these sounds, 10000 pairs in total, as indicated in Figure 3. Note that there are three clusters with close intra-cluster similarity, and they are later tagged as harmonic sounds, impulsive and impact sounds, and continuous sounds. Based on the similarity matrix, the HCA technique is applied and used to construct a heirarchical clustering of these sounds, as shown in the same figure.

A subset of the full HCA tree is depicted in Figure 3 (c). Perceptually similar sounds are closer in the tree, e.g., *typewriter* and *keyboard* sounds are under the same node and their HCA distance (tree-edge) is two units (one unit from *typewriter* to its parent node plus another one from the parent node to *keyboard*). The distance metric applies to any two sounds in the tree. The branch nodes are named to describe the meanings of a cluster of sounds that are under that particular branch; e.g., *gun* and *axe* sounds are clustered as *destructive* sounds. All right-side sounds are *single impact* sounds, and the overall tag is *impulsive* for all the sounds in this figure.

These 100 sounds provide a carefully chosen, representative set of common environmental sounds, and we leverage existing perception studies [Gygi et al. 2007] to ground our approach in human factors research. The sound duration is limited to about 1 second which is long enough for a distinct sound event; sounds with longer durations can be segmented and processed in sequence. Our framework can easily be extended to new sounds by importing raw sound data and extending the HCA tree. The clustering information can be acquired from existing studies, running new human subject experiments, or manual labeling, and is not the focus of this paper.

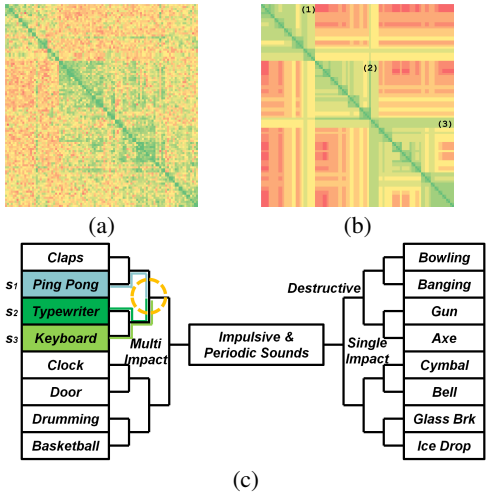


Figure 3: *Sound Perception Similarity Matrix: (a) is the sound similarity matrix constructed from human evaluations, while (b) is calculated according to the HCA tree structure. A green comparison block means that people think two perceived sounds are similar as in (a); green in (b) means the sound nodes are closer in the tree. (c) shows that the similarity matrix (a) and its clustered block counterparts marked in (b) can be transformed into a partial hiearchical clustering tree (HCA) via Multidimensional Scaling [Bonebright 2001]. Note that if multiple sounds are identified as similar to a given signal, we hypothesize that people will perceive that signal as a coarser category which is the least common ancestor (the yellow circle) of the sounds ($s1$, $s2$, and $s3$). This idea will be described in detail in the perception section.*

3.2 Sound Packet Representation (SPR)

A sound signal is traditionally represented by a wave-time or spectrum-time graph which models these three fundamental features – amplitude, frequency, and duration. SPREAD employs a packet based discretization of sound – the Sound Packet Representation (SPR) based on the Short-Time Fourier Transform (STFT) analysis technique [Hory et al. 2002] – which can be efficiently propagated using computational methods.

In Figure 4, we show that we can reduce the number of packets by using fewer frequency bands and only storing packets for sound segments with a significant amplitude. Along the horizontal axis, we represent a signal as a time-varying packet sequence. Either one packet with one amplitude value in a frequency range is generated at a time step or multiple packets for various frequencies are generated at each time step.

In SPREAD, a packet $p(i,j)$ is denoted as $\langle a, \mathbf{r} = (r_L, r_H), s\rangle$, where i is the time axis index, j is the band index along the frequency axis, a is the amplitude, r_L is the lower bound of the perceptive frequency band, r_H is the upper bound, and s the spread

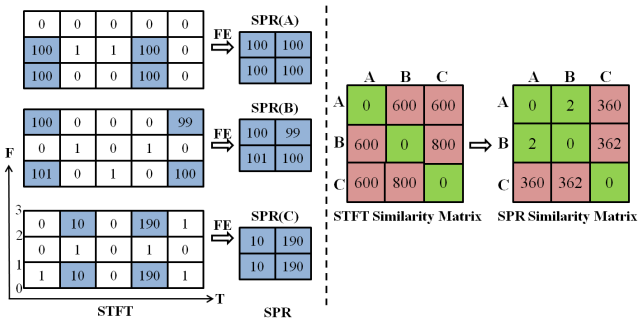


Figure 4: *Sound Packet Representation (SPR). The left diagrams illustrate the STFT conversion of the sound signal by uniformly segmenting the time (T) and frequency (F) domains, where the numerical values are the amplitudes within each discretized block. The middle (SPR) column shows the feature extraction (FE) process which determines the most distinct features among all blocks that are packed into the SPR. The right diagrams shows that we can construct different representations in order to find an optimized similarity matrix that best matches the known HCA clusters.*

factor which defines the degradation extent of the packet. Thus, a sound signal is represented as a collection of packets $\{p(i,j) \mid t_0 \leq i \leq t_n, b_0 \leq j \leq b_m\}$. The time duration of the sound signal is the total length of the packet series $t_n - t_0$ along the time axis.

In 3.3, we describe the algorithm to extract the minimal yet sufficient set of crucial and necessary packets in M frequency bands and N time slots from the original STFT data to represent a sound and the correct clustering of sound categories. Overall SPREAD efficiency is improved if unnecessary packets (relative to the selected sound database) are eliminated.

3.3 SPR Selection for Hierarchical Cluster Analysis

SPR is for a single sound whereas HCA is for multiple sounds. To establish a meaningful relation between these two we must define a comparison measure within the simulation. What we want is that sounds under the same or close clusters can be measured and evaluated as perceptually similar, while those which are more distant should be judged as different. The chosen comparison measure must operate on any subset of the sound packet data.

Dynamic Time Warping [Turetsky and Ellis 2003] is a technique to compare two time sequences, and SPREAD uses it to determine the similarity between two sounds in wave and/or STFT spectrum forms. Note that packet sequences will all have $\|N\|$ frames and within each frame there will be M packets in different frequency bands. The difference between any two frames are defined as the sum of all the corresponding packet pairs (i.e. $d(f_j, f_k) = \sum q_{mi} - r_{mi}$). The distance $DTW(s_a, s_b)$ between two sound signals s_a, s_b is computed by applying the Dynamic Time Warping algorithm to the packet sequences in s_a and s_b, as in Eq. 1:

$$DTW(s_a, s_b) = min\{C_p(s_a, s_b), p \in P^{len(s_a)\times len(s_b)}\} \quad (1)$$

where $len(s_a), len(s_b)$ are the total number of time frames of s_a, s_b respectively, $P^{len(s_a)\times len(s_b)}$ is the set of all possible warping paths in the cost matrix $d(i,j)$ and $C_p(s_a, s_b)$ is the cost of two sequences along the path p which is the min-cost frame-to-frame mappings between them from the beginning to end along the time indices. The amplitude a, frequency band range $\mathbf{r}$, and spread factor s are used to compute the metric difference between any two packets (i and j) in the sequences: $d(i,j) = \mu(a_i - a_j) + \nu(1 - \frac{\mathbf{r}_i \cap \mathbf{r}_j}{\mathbf{r}_i \cup \mathbf{r}_j}) + \xi(\frac{\|s_i - s_j\|}{\|s_i + s_j\|})$. In our problem setting, we have $\mu = 100$, $\nu = 1$, and $\xi = 1$.

Our requirements for SPR are that it: 1) be minimal yet sufficient (it needs to be the minimum representation that can sufficiently distinguish between all leaf nodes in the HCA tree); 2) should not be so fine that it wrongly discriminates similar sounds in the same category; 3) should be computationally efficient (as a smaller subset of data may be used). Thus, we seek to find optimal representational subsets of the data, using Algorithm 1, Eq. 1 and Eq. 2. In Eq. 2, t denotes a tree node, R_t is the regularization value on t, R is the total, (a,b) denotes all sub-leaf node pairs under the tree node t, D is the DTW function as defined above.

$$R = \sum_t R_t$$
$$R_t = \begin{cases} 0, & \text{if } t \text{ is a leaf node} \\ \frac{\sum_{(a,b)} DTW(t_a, t_b)}{\#of(a,b)pairs}, & \text{if } t \text{ is not a leaf node} \end{cases} \quad (2)$$

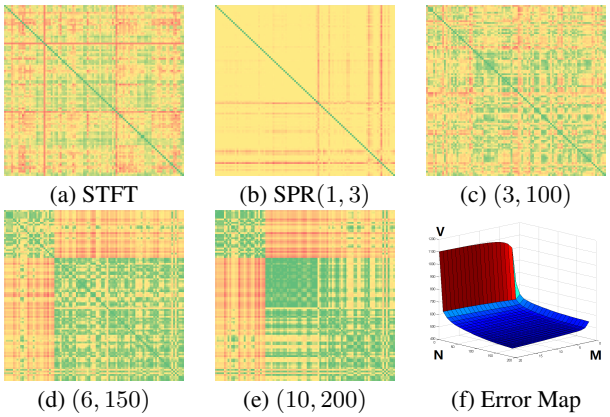
Input: Sounds $S = \{s\}$ in STFT form, HCA Similarity Matrix H
Output: Sound Packet Representation M, N
Given $M = \{M_s\}$ ($\|M_s\| \leq 10$), $N = \{N_s\}$ ($\|N_s\| \leq 200$);
1) Generate randomized sampling sets on frequency/time slots:
foreach *Sound s in S* **do**
  $M_s = (i_0 \leq i_1, ..., i_m)_s$ & $(N_s = j_0 \leq j_1, ..., j_n)_s$;
end
2) Construct **SPR**(s, M_s, N_s) for each sound in S;
3) Compute the DTW similarity matrix D on SPRs:
foreach *Sound signal pair* $(s_I, s_J) \in S \times S$ **do**
  $D(I,J) = DTW(SPR(s_I), SPR(s_J))$;
end
4) Normalize D and calculate $V = \|H - D\|_2$:
5) Iterate 1)-4) to find the best $M, N = argmin_{M,N} V + R$;
Algorithm 1: Sound Packet Representation Optimized to Match with HCA Tree. H is the HCA node-to-node tree distance matrix. R is a tree regularization term, which is defined in Eq. 2

Our SPR framework selects representative frequency and amplitude features from audio clips, but the sparse sampling may fail to capture salient differences that a person would normally perceive, while dense sampling may introduce noise and error and also fail to show distinct differences. To minimize this ambiguity we use an algorithmic feature selection process based on human sound perception. Feature selection is optimized to match the target sound perception space stored in an HCA tree structure, so that the difference between any two signals has a distance similar in scale to the corresponding two nodes of the HCA tree.

Figure 5 (a) shows that the ordering of sound signals based on HCA tree distance can differ from the ordering based on the distance computed using DTW, resulting in incorrect perception clusters of sound signals. To offset this issue, we must select features of the sound signal by sampling at specific time slots such that the computed distance aligns with the perceived difference. Algorithm 1 describes the feature selection process by choosing a set of sampling slots N, M along the time and frequency axes such that the DTW distance between all sound signals is aligned to their HCA distance.

If we compute the similarities among the complete dataset (shown in 5 (a), the result differs too much from human perception and will give unsatisfactory or implausible matches. After the optimization

and construction algorithm, the similarity is shown in Figure 5 (e) which matches well the human subjective results. The factor analysis is shown in 5 (f).



(a) STFT (b) SPR$(1,3)$ (c) $(3,100)$

(d) $(6,150)$ (e) $(10,200)$ (f) Error Map

Figure 5: *Sound Perception Similarity Matrix, in which sound-to-sound similarity is calculated by DTW on the following different datasets (M is number of frequency bands and N is number of time samples): (a) the original STFT data consists of, on average, 500 time slots and 512 frequency bins. (b) SPR data constructed from HCA using M = 1 and N = 3 making the sounds hardly distinguishable. (c) using M = 3 and N = 100. (d) using M = 6 and N = 150. (e) using M = 10 and N = 200. In (f), the curved surface image shows that if M and N are set to their maximums (10 and 200, respectively) then the similarity error V is minimized. Lower M and/or N increase error. A suitable error tolerance can be set at the application's discretion.*

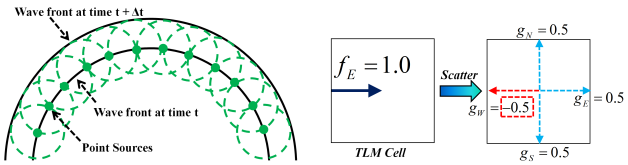
4 Sound Packet Propagation

Many sound propagation methods exist, such as FDTD and FEM in the Numerical Acoutics (NA) field, and ray-tracing and beam-tracing in the Geometry Acoutics (GA) domain. In this work, we use a rectilinear cellular space that approximates the physical environment of static and dynamic objects and agents, and propagate sound by the TLM Cell-automata Acoustics (CA) model. CA's computational cost is independent of the number of agents;GA increases per agent. Also, GA physically approximates sound waves as lights, and the GA diffraction model is expensive, whereas CA inherently models all sound/environment interaction effects.

4.1 Transmission Line Matrix Using Uniform Grids

The sound signals received by agents depend on the cell they occupy, but their other actions (such as navigation) are not restricted to this grid. Changes in a packet's feature value are governed by known formulas for sound propagation effects. For a detailed review of the TLM algorithm on uniform grids, please refer to [Kristiansen and Viggen 2010].

The TLM method belongs to the Cell-automata Acoustics (CA) category along with other methods such as Lattice-Gas and Lattice-Boltzmann models, and it is based on Huygen's principle, as shown in Figure 6 (a) & (b), that each point in the wavefront is a new source of waves. Given a grid-based discretization of the scene, the sound distribution can be calculated by: 1) updating the current energy values for each grid cell and 2) for each neighbor of each grid cell calculating the energy that will be transferred from the center grid to the neighbor grid, as shown in Equation 3. To model reflection, the vector values which reach a wall will simply reverse direction.



(a) (b)

Figure 6: *(a) Illustration of Huygens principle: starting from a spherical point source, the wavefront in the next time step is formed by propagation at the border of the current one. (b) Grid-based sound packet propagation: An original incoming packet (the energy with an arrow pointing to the center of the TLM cell) will scatter into four subpackets ($f_E \rightarrow \{g_N, g_E, g_S, g_W\}$) which are the outgoing packets to be transmitted to its neighboring four-connected cells (N,E,S,W), where they become new incoming packets in the next time step.*

TLM can simulate multiple simultaneous sound sources anywhere in the grid. Packets at the same locations and the same bands will be merged and their amplitudes will be added. The TLM grid can also represent 'constant' ambient sounds (e.g., general levels of traffic noise) that just sum with any transient packets. Such levels can be ascertained empirically or from other simulations [O'Sullivan and Ennis 2011] and create appropriate perceptual confusions. Moreover, the agents themselves can generate local sounds (*footsteps*, *handclaps*, or *non-linguistic utterances*) in the grid. They increase the grid absorption but do not otherwise impact the propagation algorithm.

$$a(g_i) = \alpha(g_i) \cdot (-\frac{a(f_{op(i)})}{2} + \sum_{j \neq op(i)} \frac{a(f_j)}{2}) \qquad (3)$$
$$a(Neighbor(g_i)) = a(g_i)$$

$$s(g_i) = \sum_{j \neq i} s(f_j) \cdot \delta_s(g_i)$$
$$\delta_s(g_i) = \begin{cases} 0.01, & \text{if } g_i \text{ is an agent grid} \\ 0.10, & \text{if } g_i \text{ is a wall grid} \\ 0.98, & \text{if } g_i \text{ is an ordinary grid} \end{cases} \qquad (4)$$

$$s(f_i') = \texttt{Collect}_s(i, \{s(g_0), s(g_1), ..., s_(g_k), ...\}) \qquad (5)$$

We extend the scatter rule in Equation 4 to work for the sound packets' spread factor. Here, $s(g_i)$ is the spread factor which indicates how clear or fresh the packet is at grid g and direction i, and $\delta(s)$ is the decrement multiplier for the factor. The collection rules in Equation 5 merges the incoming packets together with their spread factors merged (summed). These changes do not affect TLM: we just focused on propagating key packets, modeling their interactions, and tracking their degradation with the spread factor.

Note that our framework didn't fundamentally change TLM, but we propagated only key segments of (wave) packets, modeled their inter-impacts, and tracked their spreads during propagation. The spread is regarded as an factor of how much a packet has endured or degraded and how many HCA candidates are qualified.

4.2 Pre-computation for TLM using a Quad-Tree

Input: Quad tree Q
Output: Precomputed propagation values H
foreach *Possible Size z of the Grids in Q* **do**
  Let B be a uniform grid of size $z \times z$;
  foreach *Border grid $b \in B$* **do**
    foreach *Incoming $i \in \{N, S, E, W\}$* **do**
      Incoming unit energy in B at b from i;
      for $t = 1;\ t < L = t_{max};\ t++$ **do**
        $B_t = \textbf{TLM}(B_{t-1}, b, i)$;
        $H(z, b, i) = H(z, b, i) \bigcup B_t$;
      end
    end
  end
end
Sort all H values in ascending t & descending amplitude;

Algorithm 2: Pre-computation of propagation values in quad tree. Note that H is a cached set of packets for a square region of size z (which can only be 1, 2, 4, ... due to quad-tree settings), and for the case that a trigger packet is incoming at border grid b in the i direction. It stores the propagation patterns at the border grids from time t_1 to t_{max} i.e. L. The last sorting step is to reduce unnecessary checking in latter frames of the set. For example, in time t a cached value h_t is already smaller than a threshold ϵ, so checking $t+1, t+2, ...$ is no longer necessary.

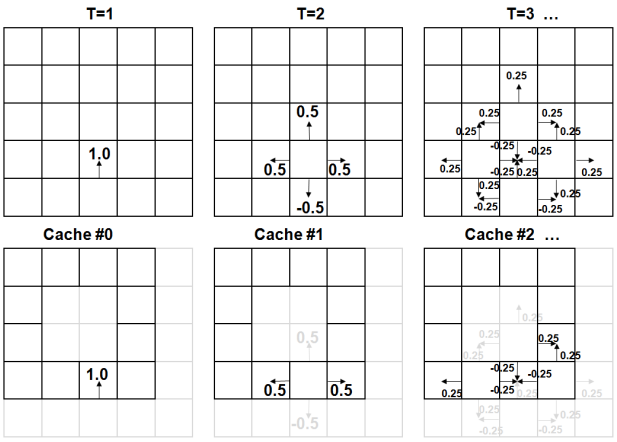


Figure 7: *Using the TLM pre-computed cache. The top row shows a number of frames for propagation in the full domain. The bottom row shows a highlight view of only the border grids of a quad region (here 4x4). For a unit incoming trigger packet, its consequent propagation pattern is deterministic so can be pre-computed and cached. For any incoming packet with value v, multiply v by the cached values and apply them to future frames.*

For a square region with the same sound attenuation property, the propagation pattern is always the same and is proportional to the original source energy, which can be pre-computed and cached as shown in Figure 7. Moreover, since the sound signals we are processing are fairly short with only about 150 frames of packets, any sound trigger only needs to be propagated for less than 3 seconds. This particular constraint allows us to cache some of the propagation results and accelerate the overall algorithm. Algorithm 2 describes the pre-computation of energy patterns in quads of different sizes.

We subdivide the entire scene into quads such that each quad region has uniform acoustic properties. Given any input sound, we can find the relevant propagation pattern and its result, and then assign the distribution values to the incident grids, repeating this process for each timestep. Algorithm 3 describes the modification of the uniform TLM (UTLM) algorithm to work in an adaptive quad-based environment using pre-computed propagation values. The propagation results using this method are identical to using a uniform grid, as illustrated in Figure 8, and provides a tremendous performance boost (Figure 9).

Input: Pre-propagation quad tree Q
Output: Post-propagation quad tree Q'
foreach *quad $q \in Q$* **do**
  if *q has existing packets* **then**
    foreach *Border Grid $b \in q$* **do**
      if *b has existing packets* **then**
        Let q' be the neighbor quad;
        Let b' of b be the neighbor grid at q';
        Let v be the `Scatter` value from b to b';
        `Collect` $v \cdot H(\|q'\|, b, i)$ on q''s borders
        (Complexity $\leq O(T * L) \ll O(R * R)$);
      end
    end
  end
end

Algorithm 3: Sound propagation in adaptive quad-based environment representation using pre-computed propagation values. R is the space resolution size, T is the number of border grids for the largest quad, and L is the largest index of the future frame that a cache will be saved to. The `Collect` step is for a set of cached packets, not a single one, and this step will introduce error if we set a truncation threshold as described in Algorithm 2. Based on our experiments, with a fairly large $\epsilon = 0.001$, the relative error is lower than 2% within an acceptable range.

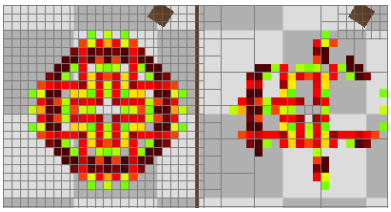


Figure 8: *TLM Quad Tree. The left diagram shows the TLM result on a uniform grid, where the red color means a high sum amplitude of all packets within the cell, and green means low sum. the right diagram shows a quad-tree grid. In the quadtree only border grids need propagation: the 'internal' grids are unnecessary because no receivers exist within that region (if they did that region would have been previously subdivided).*

To explain why QTLM gives approximately linear runtime (with regard to the log scale of space resolution as shown in the figure), UTLM on $R*R$ grids has complexity of $O(R*R)$ because it needs to update all its grids, but QTLM only needs to update the borders of grids, which is approximately $O(R)$ because only the border grids count. Then for each border grid's effect, it needs to update T (at most $4 * R$ for a quad) other border grids in L consecutive frames, $O(T * L)$ in total. Since practically a single packet won't impact most of the border grids or most of the following consecutive frames within the current one frame, $T \ll 4 \cdot R$, much less than $O(R)$ and moreover $L \leq R$, and in total $O(T * L) \ll O(R * R)$. In fact, the larger the value R is, the more runtime will be saved (because $L \ll R$ then) with the trade-off of greater (but one time) overhead of pre-computation time and more cache storage. Updates

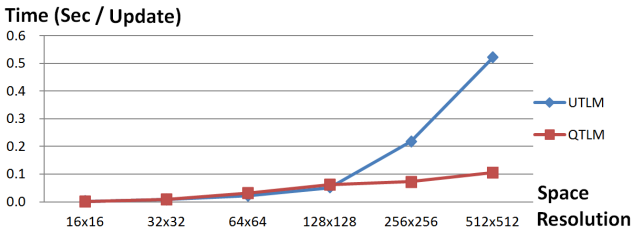


Figure 9: *Performance comparison between Quad-tree TLM and Uniform TLM. Computational cost of UTLM increases quadratically, while QTLM increases only linearly. For a 512x512 resolution, QTLM takes about 5G memory and 15 seconds overhead pre-computation on an x64 machine. But with further optimization, these constraints could be much reduced.*

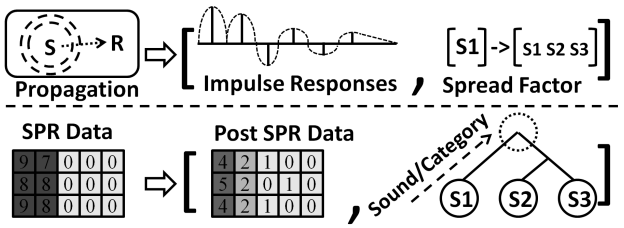


Figure 10: *Post-propagation data used for sound perception model. The top row shows that after propagation, the impulse responses (IR) of the original sound signal sequence (from SPR) will be received along with the spread factor indicating any frequency degradation. The bottom row shows how post SPR data, which is computed based on IR, resulted from degradation and change during propagation, affects the perception of sound category.*

on quads without packets are obviously unnecessary. In total, the combination gives approximately $O(R*T)$ ($O(R)$) performance, also reflected in our chart. Furthermore, quad-tree pre-computation is only for each different size of obstacle-free quad (1x1, 2x2, 4x4, 8x8,...), and the pre-computation does not need to consider any nested tree configurations. The limitation of this algorithm is that it is not suitable for long duration propagation because L will be very large and the computational and space cost will be very expensive. However, only high amplitude sounds will create large L. Based on existing algorithms [Li and Loew 1987], quad-tree updates can be achieved in $O(1)$ complexity, as long as the dynamic changes only affect neighbor grids. This fits with our dynamic simulation framework for autonomous agents.

5 Sound Perception and Behaviors

Hearing helps us experience, communicate with and react to the ambient environment and other people. Leaving aside linguistics, we can still build a sound perception model for virtual agents so they can identify, as well as possible, any environmental sound packets they receive.

5.1 Effect of Sound Degradation on Perception

Agents perceive two types of information from any packets that arrive at their ground location: 1) the impulse responses of packets at different frequency bands and 2) the spread factor that is computed for each packet which indicates the frequency and amplitude changes due to environment interactions and attenuation. Then we use Dynamic Time Warping (DTW) [Turetsky and Ellis 2003] to compute similarity values between these packets and all the sounds in the HCA database. The similarity value ranks possible leaf node matches or probable general categories related to the spread factor. The process is shown in Figure 10.

5.2 Hierarchical Sound Perception Model

Agents should be able to identify clear sounds accurately, but degradation may confound accurate identification. We exploit this to model sound perception based on the HCA tree structure. E.g., *ice drop* and *glass break* are grouped as a single *impact* sound which is *non-harmonic* and *impulsive*. *Blowing*, *gun*, and *axe* sounds are grouped together into *destructive* in their common ancestor node, and other sounds such as *clocks*, *drums*, *claps*, and *typewriter* are grouped as *multiple impact* sounds.

If an agent is unable to accurately perceive a sound (a leaf node) due to packet degradation, it may still find a similar sound type at a coarser level in the HCA tree. An unintelligible sound would map to the root node of the tree: the agent hears something but cannot identify what it is.

An agent may receive a temporal series of packets from more than one source. The packet series of each sound in the database are compared with the received series using DTW. Since there may be multiple sound sources, the first matched packets will be removed from the received packet set, so the extract and match processes can continue with the remaining ones. For example, in a series of received packets, suppose there are three distinct sets, and two of them are in the high bands and belong to the *siren* sound, then they will be used to match first. The remaining (one) low band will be used to compare with other sounds. Note that this greedy step will introduce error. Although people are good at distinguishing convolved sounds, a perfect blind source separator is difficult to model [Bee and Micheyl 2008].

Input: Post-Propagation SPR Data Pp, SPR Database B
Output: Perceived Sound Category C, Sorted Match List L, Spread Factor S
$S = \sum p.spreadFactor$;
K = max(1, (1-$\frac{S}{S_{ref}}$) $*$ T) (where $T = 20$ in current setting);
L = ComputeDTWSimilarity(P, B);
L_K = Extract Top K Candidates from L;
C = Find the Lowest Common Ancestor of M_K in the HCA Tree;
Algorithm 4: Sound Perception Algorithm.

The spread factor value models an SPR's range dispersion. The smaller the spread factor, the more degraded and approximate will be the perception. The relation between spread factor and candidate number K is chosen to be linear, though other relations could be used instead. Assume the reference spread factor is S (i.e., 10), then if any sequence's specific spread value sum (of all the received packets) is more than 95% of S, then only the top (first) candidate will be considered and used to find the HCA node; if more than 90% then the top 2 candidates, 85% the top 3, and so on.

In terms of identifying the perceived sound information, the top K (≥ 1) candidates below a similarity comparison threshold (least similarity * 10) are output as the set of perceived sounds. These sounds map to leaf nodes in the HCA tree structure, and we define their least common ancestor as the perceived sound category. As shown in Figure 3 (c) , a given sound signal has similar sounds s_1, s_2, and s_3, and so the perceived sound category is their least common ancestor. Figure 1 illustrates the perception results of the *glass break* sound at different locations: (1) open area, (2) high absorption region, (3) high reflection region, (4) blind corner, and (5)

sound blocking region. The *glass break* sound is clearly heard in nearby or open areas, with coarseness of perception increasing in complex surroundings with obstacles and other agents. In contrast, a harmonic sound like *harp* is accurately perceived in most of the areas. These examples show that our sound perception model accounts for sound characteristics and the dynamic configuration of the environment, and is not a simple distance-based perfect reception function. The process is described in Algorithm 4.

5.3 Sound Attention and Behavior Model

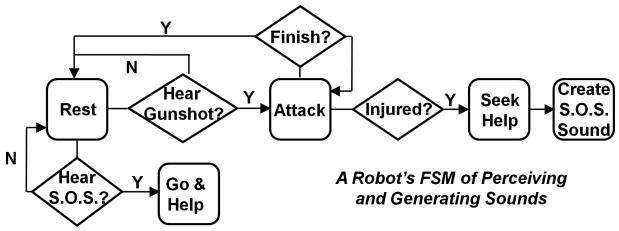


Figure 11: *Finite State Machine example for modeling a robots behavioral response to auditory triggers.*

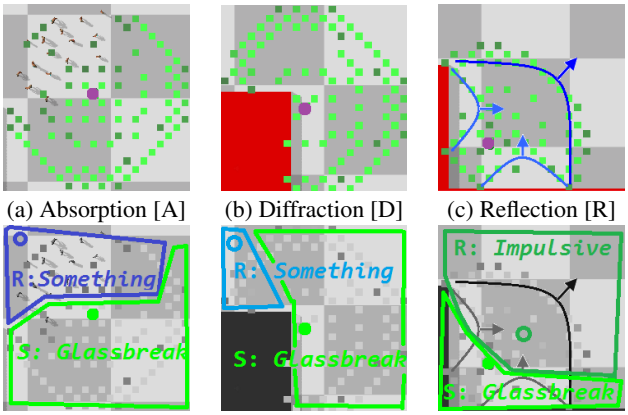
An agent's response to sound depends on being able to hear and possibly disambiguate it from noise, but it also crucially depends on a human cognitive property: attention. We use a very simple model based on two attention measures. The first is the amplitude threshold $A = 100$. When any sound's total amplitude, the sum of all packets during a number of frames ($\simeq 150$), exceeds this limit it will definitely draw one's attention. Sounds which have low amplitudes are unintelligible or just contribute to nondescript background noise. The second measure computes the saliency or conspicuity of a new sound by comparing its packets with those of the previous sounds: if the difference between them is greater than a percentage threshold $P = 30\%$, it also triggers the agent's attention. These attention triggers can be used to select, modify, or terminate associated agent behaviors. Figure 11 illustrates a simple finite-state controller for agent steering behaviors based on audio perceptions. Other agent control mechanisms are clearly possible and can be embedded in simulations or games.

6 Experiment Results

For our experiments, we use the same set of 100 environmental sound signals that were used in [Gygi et al. 2007]. We demonstrate SPREAD using a simple virtual environment with static obstacles and moving agents. Static obstacles occupy grid cells with absorption and reflection rates for sound propagation. Moving agents dynamically map their absorption and reflection rates to the grid cell they presently occupy. Our system is built on top of the ADAPT platform [Shoulson et al. 2013] which provides tools for global navigation, goal-directed collision avoidance, and full-body character animation.

6.1 Sound Propagation Experiments

Figure 12 illustrates the different acoustic properties exhibited by our sound propagation framework. All these results arise from a single omni-directional sound emission pulse at the purple point in the images. Figures 12(a) and (b) show the propagation results without absorption and with absorption due to the presence of other agents. Figure 12(c) illustrates diffraction of sound where the green automata propagates around an obstacle corner. Figure 12(d) shows sound reflection where green automata have been bounced back



(a) Absorption [A] (b) Diffraction [D] (c) Reflection [R]

(d) Perception [A] (e) Perception [D] (f) Perception [R]

Figure 12: *TLM propagation results showcasing different acoustic properties.*

from the red walls; the deep blue arrow shows the original propagation direction and the light blue arrows show the reflected directions. A result comparison is shown in Figure 13.

6.2 Sound Perception Experiments

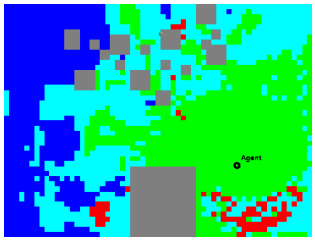
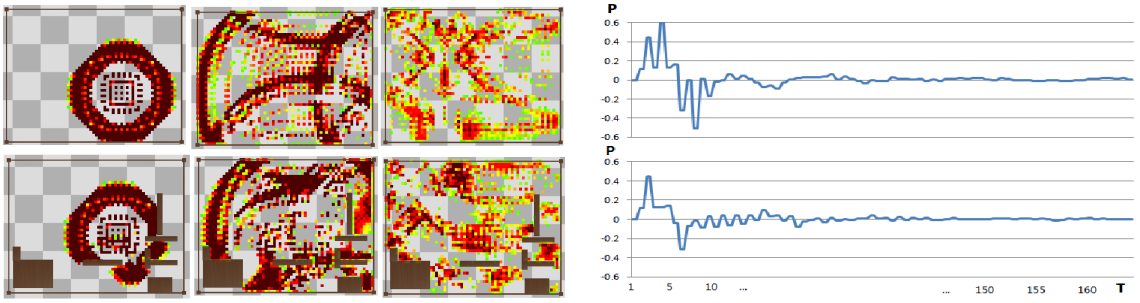


Figure 14: *Perception contour map for one agent listening to a glassbreak sound played at different locations in the environment with reflective obstacles (gray).*

Figure 14 illustrates the perception contour map using our approach. We observe a non-linear separation between regions of accurate, approximate, and incorrect perception. This is in sharp contrast to existing models which generally use simple distance-based functions, highlighting our approachs veracity.

The accuracy of sound perception with respect to the sound packet representation parameters are shown in Table 1. In the experiment, there are 30 agents, and we play the sounds at various points and capture the agents' perceptual matches. If the perceived sound is exactly the played sound, it will increase P_a which is the percentage of accurate perceptions, and if the perceived sound is an ancestor sound it will increase P_f which is percentage of approximate perception. For $M = 1$, the accuracy is very low, because adding all the spectrum values together destroys the distinct features of each frequency band. The experiments were run on a desktop PC with an Intel i7 2.8GHz CPU, 16GB RAM, and a Quadro NVS 420 graphics card. The system runs in real-time. The situation shown in Figure 1 uses parameter values: $M = 3$, $N = 50$, #agents=20, $\mu = 100$, $\nu = 1$, $\xi = 1$, effective length of sound data is typically about 1s. Currently, we can process a scene up to 150*150 grids and 50 agents in real-time, and can greatly benefit from GPU acceleration. Note that the "accurate" perception percentage is only 43.5%. Since SPREAD degrades signals, we would actually be more surprised if this were higher. This is because here we only



(a) (b)

Figure 13: *This figure shows the similar results from our TLM propagation and Raghuvanshi et al.'s FDTD method on the emission of one Gaussian impulse. Refer to Figure 6. in the work [Raghuvanshi et al. 2010]. (a) the propagation results on two different scenes of a same impulse packet; (b) the impulse response packets' values sampled along time frame at the receiver (near the source). Note that this is only for one frequency band.*

output one (the "best") result, and do not consider the spread range to find a more general category for a set of candidates. Thus even the "inaccurate" ones are still similar to the original's siblings in the HCA tree; e.g., an engine sound might degrade to a mechanical one (typically low frequency and non-periodic), but much less likely to a siren-like sound (high frequency and periodic). By considering these degraded perceptions, our algorithm gives a much higher percentage for "accurate+approximate perception".

	Acc.	$N = 10$	$N = 50$	$N = 150$
$M = 1$	P_a	0.0%	0.0%	1.0%
	P_f	22.7%	32.4%	28.7%
	P_{total}	22.7%	32.4%	29.7%
$M = 3$	P_a	19.9%	43.5%	38.0%
	P_f	1.0%	46.3%	50.9%
	P_{total}	20.9%	89.8%	88.9%
$M = 6$	P_a	36.1%	40.7%	40.7%
	P_f	32.4%	53.2%	57.8%
	P_{total}	68.5%	93.9%	98.5%

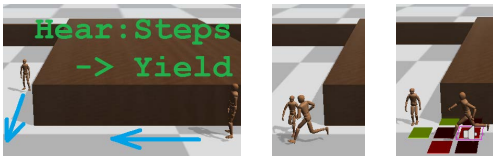
Table 1: *Sound Perception Accuracy. Accuracy statistics of sound perception for varying N and M (the number of samples along the time and frequency axes, respectively).*

6.3 Applications

We demonstrate the benefits of SPREAD by demonstrating simple applications that showcase the importance of auditory triggers in interactive virtual environments. The behavior models for the autonomous virtual humans are simple state machines which serve to showcase the significant impact agent hearing can have in simulations; they can be easily replaced by more complex behavior architectures [Kapadia et al. 2011].

Localization and Action Model. In the simulation, the sound energy distribution is calculated by summing up all the neighboring sound packets' values at all grids. Based on the distribution information, e.g. contours, gradients, etc., agent groups can navigate to different sound energy zones in the same map; e.g., some could navigate toward (or away from) different sound energy zones, others could navigate to (or from) zones with higher (or lower) energies. The supplementary video includes examples (e.g. Figure 15) where sound triggers can be used to attract the attention of other agents, or mapped to directional commands to herd a crowd.

Game Application. We demonstrate the benefit of SPREAD by integrating it into a game application, shown in the supplementary video. The original game without sound perception involves a player controlled avatar searching for and destroying enemy robots in a maze-like environment. The ability to perceive sounds greatly enriches even a simple game mechanic where robots perceive and react to different sound signals in the environment. Robots hear a gunshot and retreat or attack depending on their health status. They can additionally cry out for the assistance of nearby robots, by triggering a sound signal. Players can also mimic the robot cry to lure robots to an isolated location. The resulting gameplay is greatly diversified where players use a stealth based mechanic to isolate and corner robots in cordoned off areas where other robots are unable to see and *hear* them, as shown in Figure 11.



(a) (b) (c)

Figure 15: *Blind Corner Reaction Example. (a) Agents walking toward each other at a blind corner. (b) Without sound propagation or perception modeled, agents bump into each other. (c) With correct models, one agent can perceive footstep sounds and yield.*

7 Conclusion

This paper integrates sound propagation and human-like perception into virtual human simulations. While sound propagation and synthesis have been explored in computer graphics, and there exist extensive studies on auditory perception in psychology, ours is the first work to enable virtual humans to plausibly hear, listen, and react to auditory triggers. To achieve this goal, we have developed a minimal, yet sufficient sound representation that captures the acoustic features necessary for human perception, designed an efficient sound propagation framework that accurately simulates sound degradation, and used hierarchical clustering analysis to model approximate human-like perceptions using sound categories.

The results described in the paper and shown in the video are created using the environmental sounds provided in [Gygi et al. 2007]. Additional sound types can be added to SPREAD either by running analogous perceptual experiments or by manual annotation and placement in the HCA tree. For instance, the hierarchical clustering of phonemes [Dekel et al. 2005] could be used as the basis for the propagation and perception of speech signals.

Our method is not intended for auralization [Savioja et al. 1999], but serves as a companion to auralization – enabling virtual agents to hear and classify sounds much like their real human counterparts. Auralization is not a pre-requisite for this capability. Moreover, we need to compare the simplified SPR method with other forms of data representations for different types of sounds, as described in [Cowling and Sitte 2003]. There are no technical barriers to extending the TLM algorithm into 3D but we have no obvious reason to do so for the envisioned environments and situations.

Acknowledgements

The authors thank Alexander Shoulson for the ADAPT system, and Brian Gygi and the Hollywood Edge company for providing the environmental sound data. This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement # W911NF-10-2-0016. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

BEE, M., AND MICHEYL, C. 2008. The cocktail party problem: What is it? how can it be solved? and why should animal behaviorists study it? *J. of Comparative Psychology 122*, 3, 235.

BONEBRIGHT, T. 2001. Perceptual structure of everyday sounds: A multidimensional scaling approach. In *Proc. of the 7th international conference on auditory display*, 73–78.

CHANDAK, A., LAUTERBACH, C., TAYLOR, M., REN, Z., AND MANOCHA, D. 2008. Ad-frustum: Adaptive frustum tracing for interactive sound propagation. *IEEE TVCG 14*, 6.

CONY, C., DE LIMA BICHO, A., JUNG, C., MAGALHAES, L., AND MUSSE, S. 2007. A perceptive model for virtual agents in crowds. In *CGI*, vol. 1, 141–150.

COWLING, M., AND SITTE, R. 2003. Comparison of techniques for environmental sound recognition. *Pattern Recognition Letters 24*, 15, 2895–2907.

DEKEL, O., KESHET, J., AND SINGER, Y. 2005. An online algorithm for hierarchical phoneme classification. In *MLMI*, 146–158.

FUNKHOUSER, T., CARLBOM, I., ELKO, G., PINGALI, G., SONDHI, M., AND WEST, J. 1998. A beam tracing approach to acoustic modeling for interactive virtual environments. In *SIGGRAPH*, ACM, 21–32.

GYGI, B., KIDD, G., AND WATSON, C. 2007. Similarity and categorization of environmental sounds. *Attention, Perception, & Psychophysics 69*, 6, 839–855.

HERRERO, P., AND DE ANTONIO, A. 2003. Introducing human-like hearing perception in intelligent virtual agents. In *AAMAS*, ACM, 733–740.

HOLLAND, J., DABELSTEEN, T., PEDERSEN, S., AND LARSEN, O. 1998. Degradation of wren troglodytes troglodytes song: implications for information transfer and ranging. *J. of the Acoustical Society of America 103*, 2154.

HORY, C., MARTIN, N., AND CHEHIKIAN, A. 2002. Spectrogram segmentation by means of statistical features for non-stationary signal interpretation. *Signal Processing, IEEE Transactions on 50*, 12, 2915–2925.

JAMES, D., BARBIČ, J., AND PAI, D. 2006. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. In *ACM TOG*.

KAGAWA, Y., TSUCHIYA, T., FUJII, B., AND FUJIOKA, K. 1998. Discrete huygens' model approach to sound wave propagation. *J. of Sound and Vibration 218*, 3, 419–444.

KAPADIA, M., SINGH, S., REINMAN, G., AND FALOUTSOS, P. 2011. A Behavior-Authoring Framework for Multiactor Simulations. *Computer Graphics & Applications, IEEE 31*, 6, 45 –55.

KRISTIANSEN, U., AND VIGGEN. 2010. Computational methods in acoustics. *Compendium, NTNU*.

LI, S., AND LOEW, M. 1987. Adjacency detection using quadcodes. *Communications of the ACM 30*, 7, 627–631.

MAST, T. 2000. Empirical relationships between acoustic parameters in human soft tissues. *Acoustics Research Letters Online 1*, 2, 37–42.

MONZANI, J., AND THALMANN, D. 2000. A sound propagation model for interagents communication. In *Virtual Worlds*, Springer, 135–146.

ONDŘEJ, J., PETTRÉ, J., OLIVIER, A., AND DONIKIAN, S. 2010. A synthetic-vision based steering approach for crowd simulation. *ACM TOG 29*, 4, 123.

O'SULLIVAN, C., AND ENNIS, C. 2011. Metropolis: multisensory simulation of a populated city. In *Proc. Intl. Conf. on Games and Virtual Worlds for Serious Applications*, IEEE Computer Society, 1–7.

PELECHANO, N., ALLBECK, J., AND BADLER, N. 2008. Virtual crowds: Methods, simulation, and control. *Synthesis Lectures on Computer Graphics and Animation 3*, 1, 1–176.

RAGHUVANSHI, N., NARAIN, R., AND LIN, M. 2009. Efficient and accurate sound propagation using adaptive rectangular decomposition. *IEEE TVCG 15*, 5, 789–801.

RAGHUVANSHI, N., SNYDER, J., MEHRA, R., LIN, M., AND GOVINDARAJU, N. 2010. Precomputed wave simulation for real-time sound propagation of dynamic sources in complex scenes. *ACM Transactions on Graphics (TOG) 29*, 4, 68.

SAVIOJA, L., HUOPANIEMI, J., LOKKI, T., AND VAANANEN, R. 1999. Creating interactive virtual acoustic environments. *J. of the Audio*.

SHOULSON, A., MARSHAK, N., KAPADIA, M., AND BADLER, N. I. 2013. ADAPT: the agent development and prototyping testbed. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D, 9–18.

TAKALA, T., AND HAHN, J. 1992. Sound rendering. In *ACM SIGGRAPH Computer Graphics*, vol. 26, ACM, 211–220.

THALMANN, D. 2007. *Crowd simulation*. Wiley Online Library.

TURETSKY, R., AND ELLIS, D. 2003. Ground-truth transcriptions of real music from force-aligned midi syntheses. *ISMIR 2003*, 135–141.

UNITY3D. 2012. Unity3d game engine. *http://unity3d.com*.

XU, C., MADDAGE, N. C., AND SHAO, X. 2005. Automatic music classification and summarization. *Speech and Audio Processing, IEEE Transactions on 13*, 3, 441–450.