Sound Localization and Multi-Modal Steering for Autonomous Virtual Agents Supplementary Document

1 Alternate Model for Kalman Filter

The accuracies of direction and distance are very different so it's might be helpful to treat angular and tangential velocity independently, which means we could choose the following motion model in Kalman Filter:

$$X_t = \begin{pmatrix} \theta \\ r \\ \omega \\ v_r \end{pmatrix} \qquad A_t = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Z_t = \begin{pmatrix} \Omega \\ V_r \end{pmatrix} \qquad C_t = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

which correspond to the two major factors in human's perception of collision avoidance.

2 Possible Simplifications

Due to the cost of sound propagation, computing Kalman Filters for each source-agent pair, and maintaining an individual perception map for each agent, our framework can handle a small number of agents at interactive rates (see Results section for performance details). However, there are possible methods to simplify our framework, making reasonable approximate and thus reducing computational cost.

Simplification of Kalman Filter. If we simply use a naive model, where motion is considered together with noise.

$$X_t = \begin{pmatrix} x \\ y \end{pmatrix} \qquad A_t = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$Z_t = \begin{pmatrix} Z_x \\ Z_y \end{pmatrix} \qquad C_t = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

In this case, the Kalman Filter will converge to a simple linear combination of current and past estimates of source. This model has a noticeable time delay about estimate since it has no assumption about motion. The drawback of this simplification is that velocity is no longer a build-in component and needs additional calculation.

A simplified approach of sound-driven steering. In most applications like computer games, it is not necessary to simulate the whole process of sound propagation. One solution is to constrain sound in the certain region that we are interested. Another solution is to only simulate the wave frontier. In our model, the agent only uses wave frontier (first arrived sound packets) for localization. Since the first arrived packets propagate along the shortest path, the Dijkstra algorithm or A-star algorithm will give the shortest path. According to Huygens principle, wave-frontier approach will localize the last turning point in the shortest path as the estimated source position. Using this method, only a pathfinding algorithm is executed and in general it will give a similar result with the approach in which sound is actually propagated. One can also employ K shortest path algorithm, which allows calculating localization confidence by comparing the similarity of these paths. Let d be the length of the shortest path from the sound source to the agent, and v be the speed of the sound wave. The time delay of the sound reaching the receiver is $delay = \frac{d}{v}$. The Doppler effect can be also be integrated as follows:

$$f_r = \frac{v - v_s cos(\theta_s)}{v - v_t cos(\theta_t)} f_s \tag{1}$$

where v, v_s and v_t are the speeds of sound wave, sound source and agent, respectively. θ_s is the angle between the shortest path and source velocity, and θ_t is the angle between the shortest path and receptor.