<u>Olfactory Search in Turbulent Environments with Multiple Odor Sources</u>

Andrew Burns, Jonathan Kroc, Antony Pearson and Alexia Tatem

Introduction:

On small scales (the scale of bacteria or cells, for instance), a diffusing molecule creates a uniform gradient that can be followed by a seeker to locate the source. Such a method of locating the source of a molecule is called chemotaxis, and it simply involves the seeker moving along the path of the greatest increase in concentration. On larger scales (the scale of moths, butterflies, bees, or microrobots), the fluid environment is air rather than the liquid environment in which bacteria and cells operate. As a result, the Reynolds number becomes large. The Reynolds number is defined $Re = \rho vL/\mu$, where ρ is the density of fluid, v is the velocity of the particles in the flow, L is the characteristic length of the system, and μ is the viscosity of the fluid. In effect, this means that the trajectories of molecules carried in the flow can be perturbed and thus, chemotaxis would be an ineffective method for locating a source in such a turbulent environment. Balkovsky and Shraiman outline “a more complex strategy involving, in addition to the sense of smell, the ability to determine wind direction.” A strategy for locating the source of an odor in such turbulent flows would be useful for the design of small smelling robots used to find small gas leaks or explosives.

Methods:

Balkovsky and Shraiman proposed a discrete model of a turbulent plume. Instead of equations that describe a diffusion of particles superimposed on a wind velocity, consider an xy-plane with the source located at the origin. The mean wind velocity, *V*, we will consider to be in one direction (positive y direction) and constant in magnitude for a long time. Each time step, 3 things happen: (1) a new particle is released from the source, (2) each particle is advected by the mean velocity and moves 1 unit in the +y direction, and (3) each particle moves either -1, 0, or 1 in the +x direction, seen in figure one.

$$D = \frac{p_R + p_L}{2}$$

eq. 1

$$p(\mathbf{r}) = \frac{1}{\sqrt{4\pi D y}} \exp\left[-\frac{x^2}{4Dy}\right]$$

eq. 2

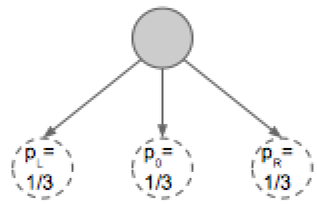

fig. 1

At scale lengths larger than *L*, the motion is Brownian, as the x-direction motion reflects the diffusion of particles. After this system has been allowed to fully develop (t>>1, with particles extending y>>1), the distribution of the particles through stochastic simulation closely matches equation 1 where equation 2 is the diffusivity coefficient. The probability distribution function given here is an analytical solution to the diffusion equation, suggesting that the discrete model is adequate to describe the mechanics of the turbulent flow. Notice also that in the fully developed plume (y>>1), at fixed y the probability distribution for a particle with respect to x-position is Gaussian.

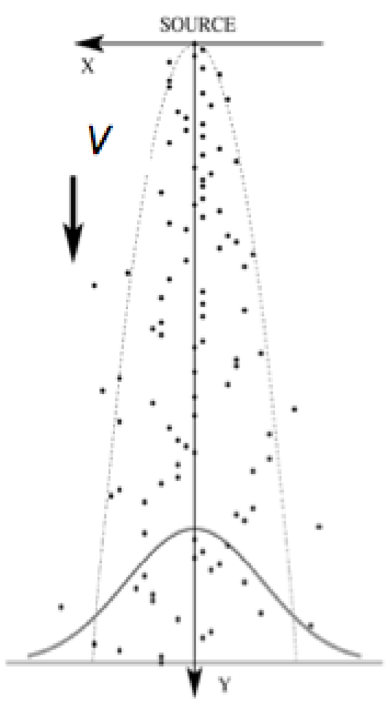

fig. 2

Consider a robot or moth located at a distance y_0 from the source. The moth can detect both the event of an odor patch arriving at its current location and the direction from which the odor patch arrived. Each time step the robot is able to move at most one lattice step along the y-axis and/or one step along the x-axis. Finally, the robot does not begin its search until it initially encounters a patch.

If (x_0, y_0) is the source of the odor patch one time step ago, the source can only be located in the interior of the cone formed by:

$$y - y_0 = \pm (x - x_0),\ y < y_0$$

This is known as the causality cone.

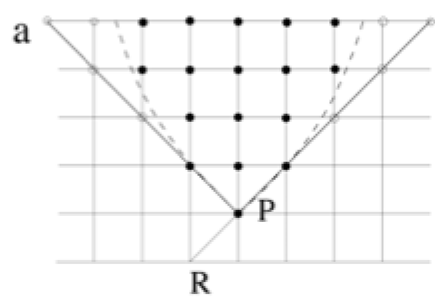

fig. 3

Multiple search algorithms are introduced by Balkovsky and Shraiman, and they are analyzed by the time it took to find the source- or the mean search time. Due to the random nature of the plume, the search time is a random quantity. Moreover, the researchers evaluated the three algorithms wand plotted the probability that the source is found during a *t*, *t*+1 interval as a function of time, ρ(t). Algorithms with means closer to zero were deemed more effective.

The first search algorithm introduced is the passive search. The robot waits at one site until it detects an odor patch. When such a patch impinges upon the robot, the robot moves along the lattice to the site from which the patch came. Over a sufficiently long period of time, this method guarantees that the robot will find the source. A major shortcoming with this method is its inefficient use of search time, a problem of particular concern in cases where the robot begins its search in regions of low odor patch density, i.e. away from the center of the plume along the y-axis. Accordingly, the authors pursued search algorithms that made better use of search time by actively seeking odor patches.

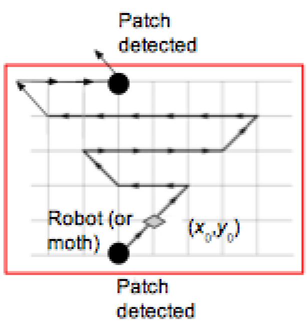

fig. 4

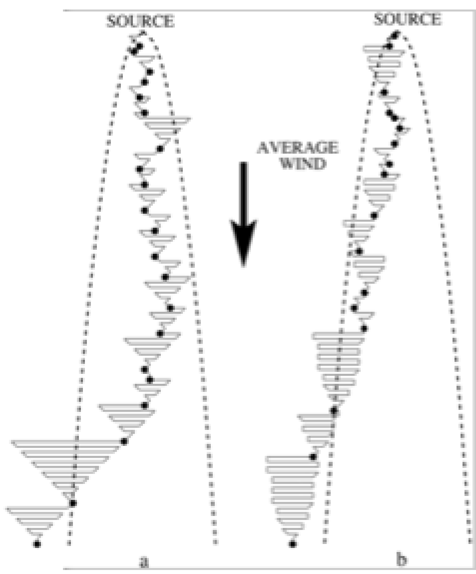

fig. 5

The first active search algorithm introduced was the conical search algorithm. With this algorithm, the robot, positioned downwind from the odor source, waits until it detects an odor patch. When an odor patch is detected, the robot moves upwind to the position the odor patch once occupied and begins its a conical search, moving along a counter-turning path, as seen in Figures 4 and 5, so as to completely cover the detected odor patch's possible previous locations. The conical search, unlike the passive search, actively seeks out odor patches. Moreover, because its path covers the odor particles causality cone, it is certain to find the odor source. Still, by covering so much area, the conical search method needlessly spends time in areas of the causality cone that are unlikely to have been the previous location of the odor particle. Accordingly, the authors next present a modified version of this conical search algorithm called the “parabolic search algorithm.”

$$(x - x_i)^2 \leq \frac{4}{3}(y_i - y)\ln\left(\frac{1}{0.05\sqrt{\frac{4}{3}\pi(y_i - y)}}\right) \quad eq.3$$

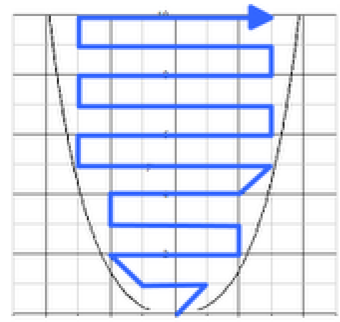

fig. 6

Whereas the conical search algorithm visited the upper “corners” of the causality cone, the parabolic algorithm passes them over. This results in a parabolic-like search trajectory, as shown in figure 6. Derived from *eq. 2*, we can see that allowing a fixed probability of an odor source originating outside our region gives an approximate parabola (*eq. 3*). In our simulations, we allowed a 5% probability of the source originating outside our parabola defined in *eq. 3*. By eliminating locations with probabilities in the tails of the distribution, the search time is thereby reduced.

To assess the algorithms, a probability distribution function of the search time is used and efficiency is defined with respect to mean search time. Balkovsky and Shraiman plotted the probability that the source is found during an interval (t, t+1) as a function of time, and algorithms with means closer to zero were deemed more effective. With respect to this definition, the parabolic search algorithm is the most effective of the three algorithms presented.

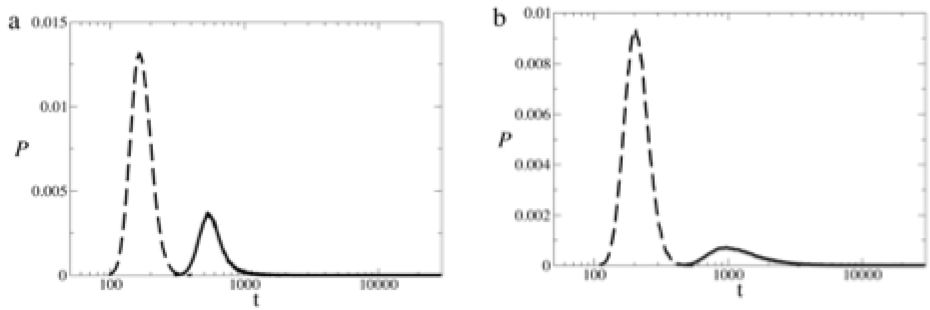

fig. 7

The histograms illustrated above, obtained via Monte Carlo simulations, model the search time. Figure 7(a) shows the robot with an initial position at (0, 50), while in figure 7(b) it is initially at (10, 50). The mean search time of the active algorithm, represented by the broken line, is not affected by adjustment of the initial position. However, the passive algorithm, represented by the solid line, is.

Project Description:

Many real world scenarios in turbulent environments involve multiple sources. Therefore, it is practical to study the impact of adding another source to our model and evaluating the effectiveness of the parabolic search algorithm. Moreover, if we found the active search algorithm to be effective, it would be useful for instances such as the design of small robots to find gas leaks or explosives.

$$P(\text{particle at x} \mid \text{fixed y} >> 1, p_R = p_L = 1/3, \text{sources at (-10,0) and (10,0)})$$
$$= 1 - \left(1 - \frac{1}{\sqrt{\frac{4}{3}\pi y}} e^{-\frac{3(x-10)^2}{4y}}\right)\left(1 - \frac{1}{\sqrt{\frac{4}{3}\pi y}} e^{-\frac{3(x+10)^2}{4y}}\right)$$

eq. 4

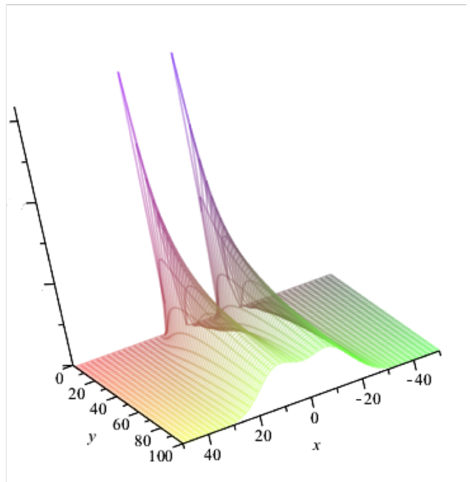

fig. 8

Figure 8 shows the probability distribution of the particles for two sources located at (-10,0) and (10,0). Equation 4 is derived from the diffusion equation, and describes the probability distribution.

First off, the parabolic search algorithm was reproduced using the software Python. Then, simulations were performed using five different initial conditions. Histograms were created for each simulation and the percentage of misses by the seeker was calculated for each run.

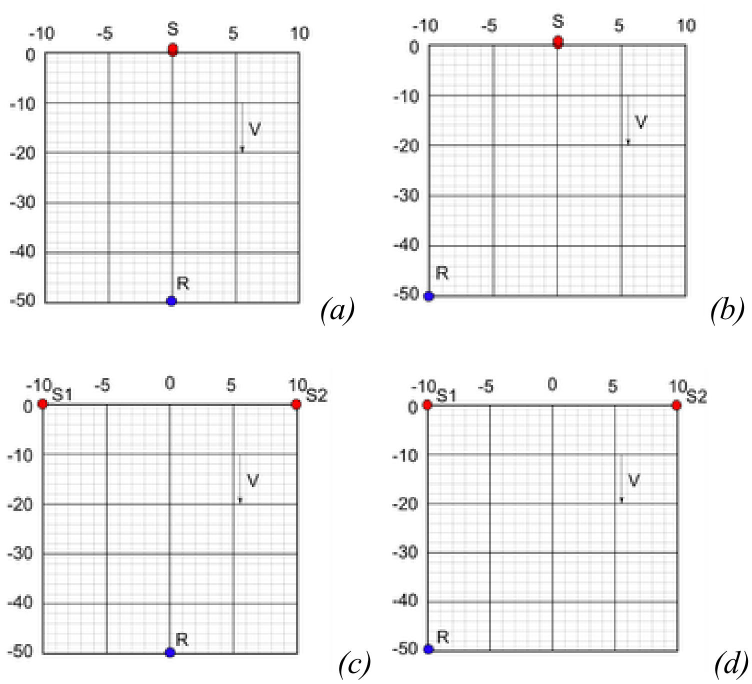

fig 9 (a)-(d)

In the first two simulations, both sources were located at (0,0). First, the robot was initially at (0,50), as shown in figure 9(a). In the second run, the robot was initially placed at (10,50), as shown in figure 9(b). For the third and fourth simulations, the two sources were placed at (-10,0) and (10,0). For the third run the robot was initially at (0,50), and for the fourth run it was at (10,50), as shown in figures 9 (c) and (d) respectively. The fifth run was a simple case with one source at the origin and the robot initially at (0,50).

The percentage of misses by the seeker, calculated for each simulation, is given in figure 10 below. A miss is counted when the seeker does not find a source. Therefore, if a seeker finds one of the two sources, it does not count as a miss.

Simulation	Miss rate
(A) One source at (0,0); robot initially at (0,50)	5282 out of 100,000 5.282% misses
(B) Both sources at (0,0); robot initially at (0,50)	2588 out of 100,000 2.588% misses
(C) Both sources at (0,0); robot initially at (10,50)	10392 out of 100,000 10.392% misses
(D) Sources at (-10,0) and (10,0); robot initially at (0,50)	12160 out of 100,000 12.16% misses
(E) Sources at (-10,0) and (10,0); robot initially at (10,50)	5318 out of 100,000 5.318% misses

fig. 10

First off, the percentage of misses by the seeker decreased when two sources were placed at the origin rather than one. This is logical, as the density of particles moving toward the robot was greater. Additionally, the miss rate increased when the initial position of the robot was offset to (10,50), as in simulation (C). This makes sense, because the robot is less likely to encounter a patch when it is not directly below the source or in a high density region. The miss rate also drastically increased when the sources were separated and the robot was placed in the center, as in simulation (D). This could be caused by a high density region of odor particles between the sources, which led the robot up the middle and caused it to miss both sources. It is worth noting that the miss rate for simulation (E), with the sources separated and the robot offset, is very similar to the miss rate of simulation (A). This is because it is essentially the same setup, with an additional source on the side. Therefore, we see that the source located at (-10,0) in simulation (E) had little to no effect on the seeker.

The percentage of miss rates by the robot in each simulation show that a seeker in a turbulent environment is less likely to locate a source using the parabolic search algorithm when there are multiple sources in different locations. Therefore, our active search algorithm is not effective for multiple sources.

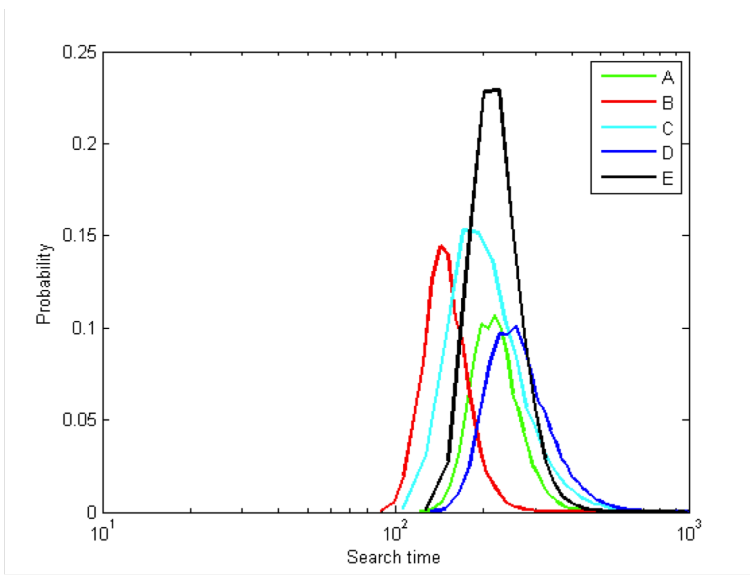

fig. 11

Figure 11 shows the histograms of search time distribution for each of the five simulations. There is a notable relationship between higher miss rates seen on the table on the previous page and wider distributions on the histograms shown here. Clearly, when the robot is initially between two separate sources the average search time is greater and there is greater variability in search times than when the sources are together. Furthermore, when one of two separate sources is directly in front of the robot initially as in simulation (A), the search times are much longer than when two superimposed sources are directly in front of the robot initially as in simulation (B). This is to be expected from halving the packet density. Simulations (C) and (D) are similar in the variability of search times with both sources offset from the initial position of the seeker, but search times are generally faster with the sources overlaid as in (C) than in (D) where the sources are on either side of the seeker.

Potential Applications and Future Work:

Our rationale for considering multiple sources is the observation that it is a common natural occurrence for multiple sources to be influenced in the same flow. For instance, consider a bee trying to land on a flower contained in a row of flowers. Moreover, an algorithm for locating multiple sources in turbulent flows would be useful for the design of robots to find gas leaks or explosives.

There are several directions that one could go when conducting future work. First off, the average trajectory of the robot for each simulation should be plotted to better understand the results. This was attempted but proved difficult to compute, given computational constraints. Also, it would be beneficial to analytically describe the search time with multiple sources.

One way to further analyze turbulent environments could be to introduce a group of entities searching for a source, as opposed to just a single seeker. Then, when one robot encounters an odor patch, the others change the angle of their search pattern. Perhaps given multiple robots, we can allow the miss rate for each robot to be quite high while still having a very low probability of missing the source altogether, but getting a much quicker search time. Another option is to create a design that alters the algorithm based upon the time between detection of the odor patches. The idea is to cover more ground where the probability of detecting an odor patch is low, and explore less area as odor patches are detected with more frequency. Finally, one could also analyze the effect of changing the width between the two sources, and find a particular source separation distance that dramatically increases the number of misses.

References:

1. Balkovsky E. and Shraiman, B.I. "Olfactory search at high Reynolds number", Proc Natl Acad Sci USA.99,12589-93 (2002).