Olfactory Search of Moths in a City Grid

Cristina Retamoza
Craig Thompson
Laura Nakolan
Noah Hammer
Steven Smith

The Paper:

Olfactory Search at High Reynold's Number

By: Eugene Balkovsky and Boris I. Shraiman

Olfactory search at high Reynold's number

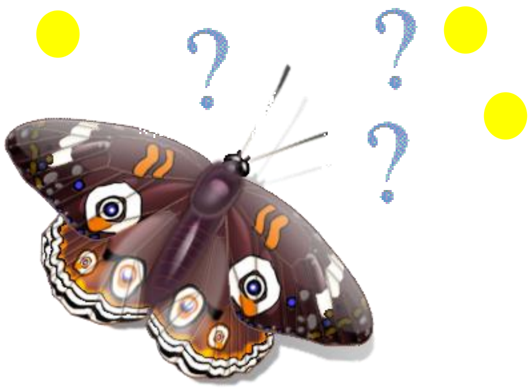
- Locating the source of odor in a turbulent environment

Animal behavior

Search for chemical leaks, drugs, and explosives

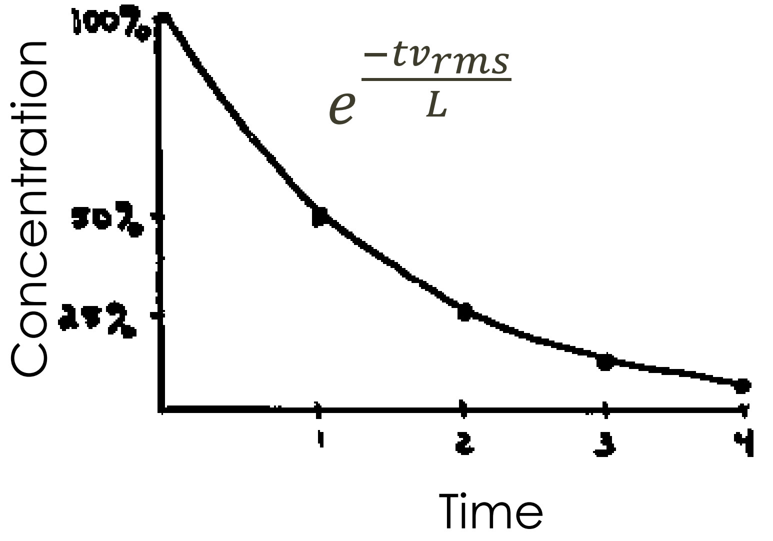
Difficulties with Turbulent Flows

- Turbulent flows consider variations at all scales
  - Large scale and small scale eddies mix
  - Diffusion
- Fluctuating structure of odor plume
  - Odor is not always present or does not point towards source
  - Concentration decreases as it moves away from the source and time between detecting odor increases.



The Odor Plume

- Inhomogeneity of velocity causes stretching of the plume and diffusion, which decays the odor
- *The probability of an odor patch to survive for time t is:*

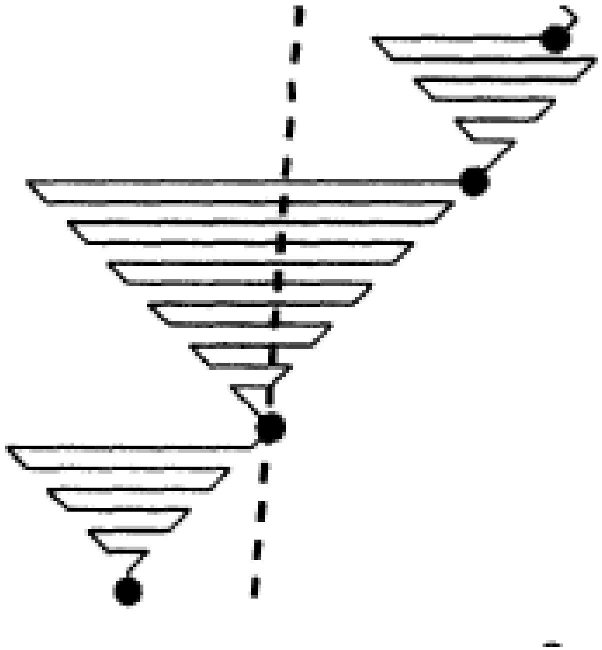


$$e^{\frac{-tv_{rms}}{L}}$$

**(vrms is the root-mean -square of the velocity)*

Moth's Strategy

- Because the female moth produces so few molecules, there is no gradient for the moth to follow
- Counterturning pattern
  - Casting—wider crosswind excursion
  - Zigzagging—upwind progression

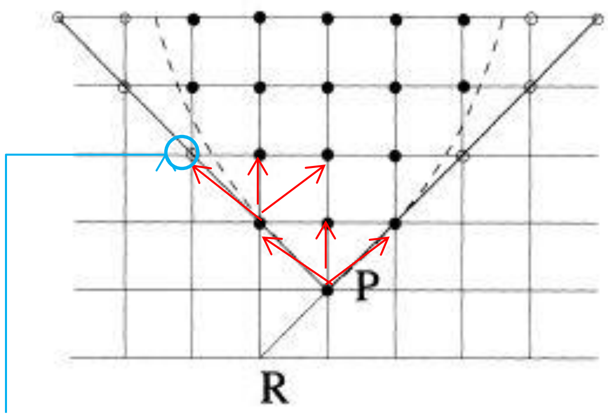


Parameters

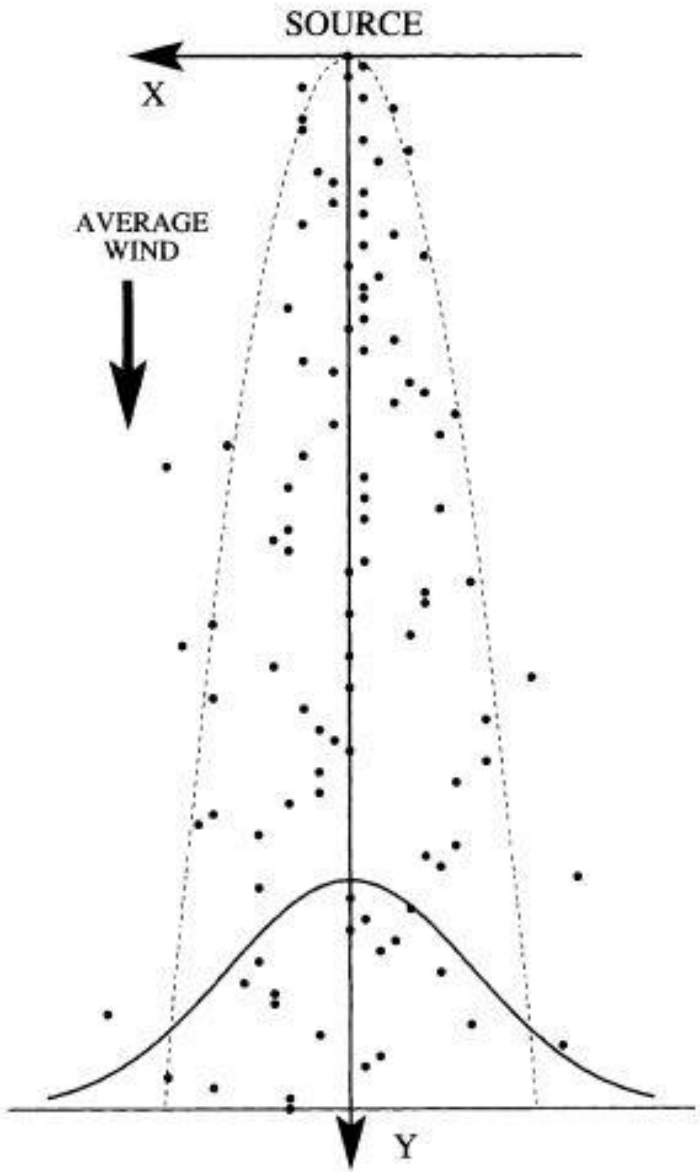
- V - Mean velocity of the wind, is set by atmospheric conditions and doesn't change.
- Odor molecules move in a random walk superimposed on the wind.
- L – height above the ground, at scales larger than L, the motion is Brownian with a diffusion coefficient estimated at Lv_{rms}
- V_{rms} - root mean square of the random walk fluctuations.

2D Model

- An odor patch that starts at (0,0) is advected to (-1,1), (0,1) or (1,1)
- (-1, 1) can move to (-2, 2), (-1, 2) or (0, 2), etc
- Probability to moving left, forward or right is equal



Statistically unlikely



2D Model

Stationary distribution of the patches

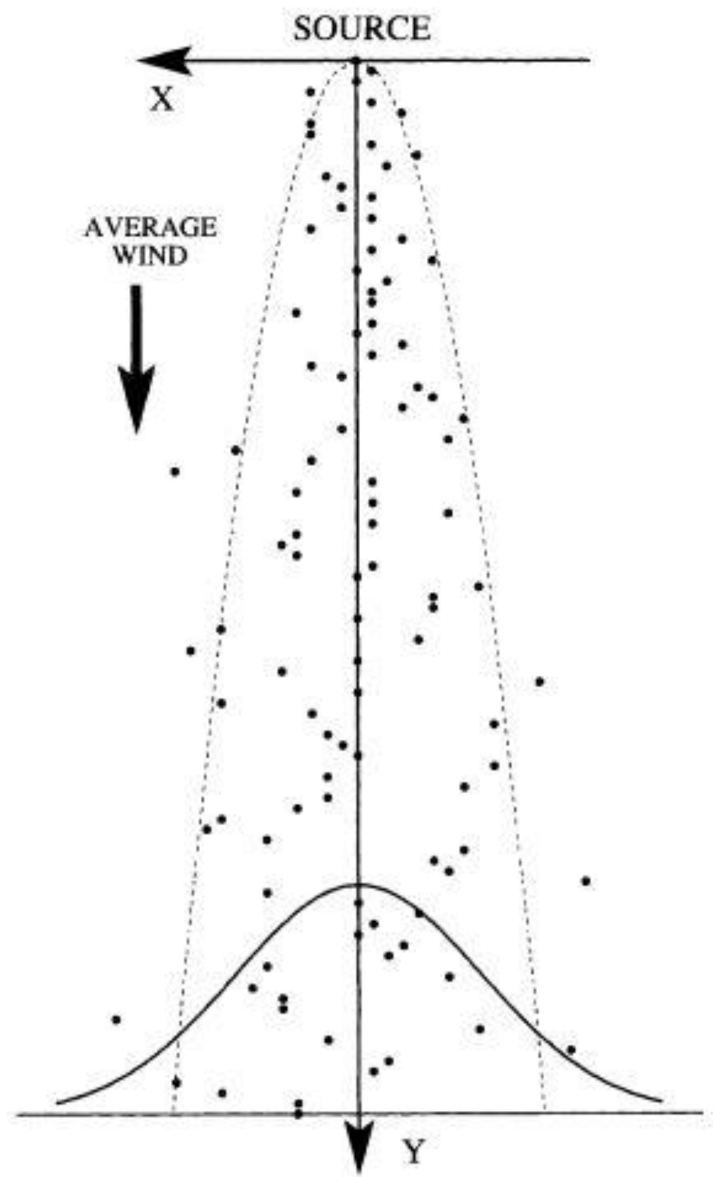
$$p(r) = \frac{1}{\sqrt{4\pi D y}} \exp\left[-\frac{x^2}{4Dy}\right]$$

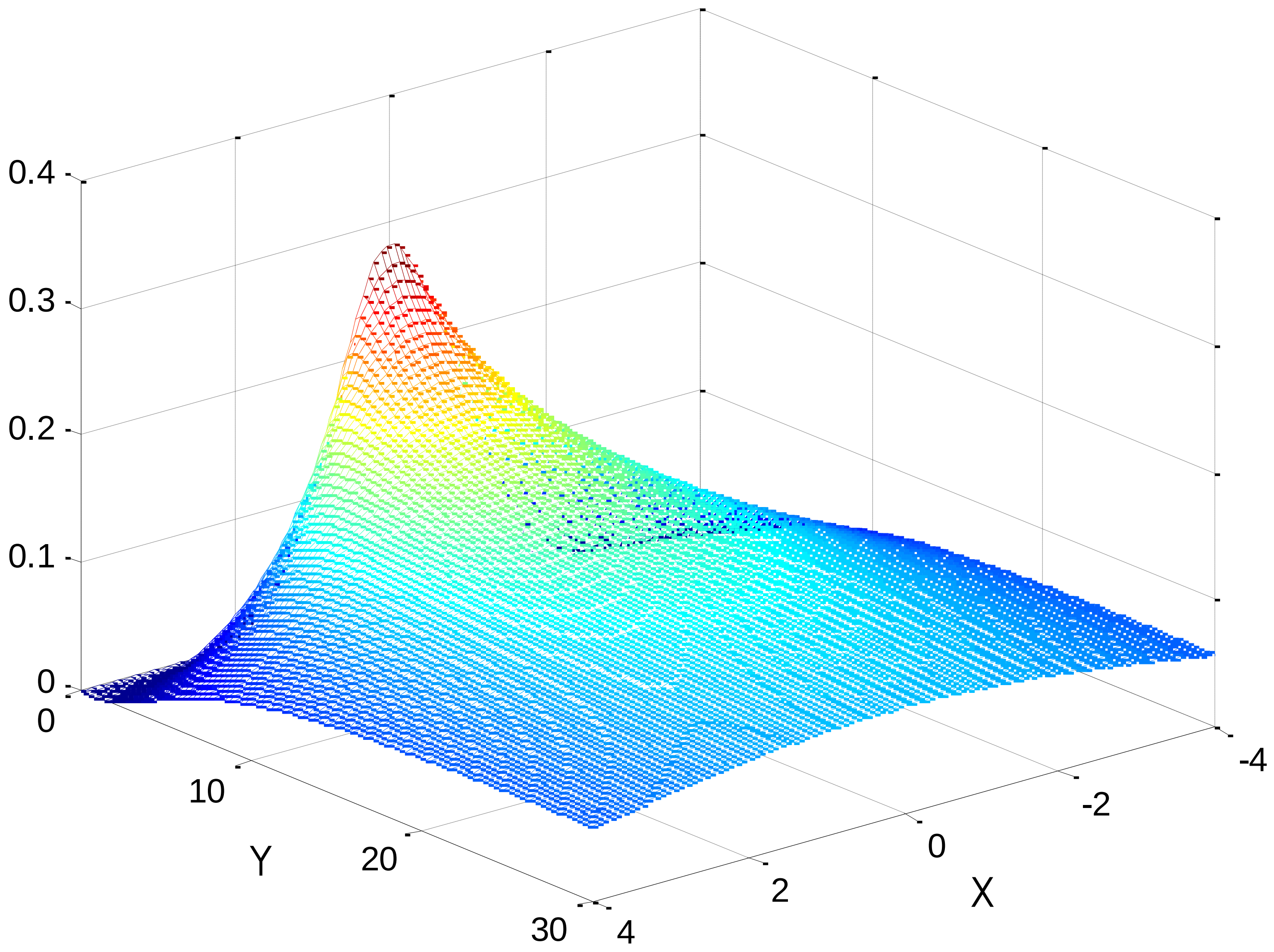
r is the point (x, y)

D is $(p_R+p_L)/2$

p_R is the probability of moving to the right

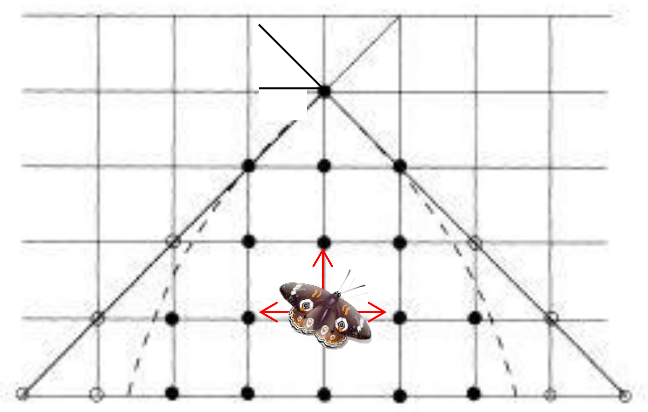
p_L is the probability of moving to the left




0.4
0.3
0.2
0.1
0
0
10
Y
20
30
4
2
0
X
-2
-4

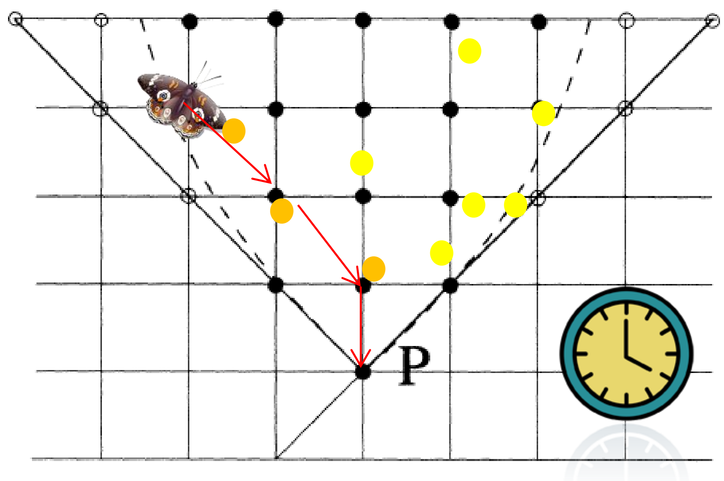
Rules for the Model

- Moth starts moving once it gets first whiff of odor
- Each time step, moth can move once
  - 1 lattice step along the x or y axis
- Moth does not move downwind



Strategy 1

- Moth waits at a sight until it gets an odor patch
- Moves to the site from which the patch came
- 100% probability of finding source, but….
  Not most efficient



Strategy 1

The Probability Distribution Function (PDF) of the "passive search" time is:

$$\rho(t) = \frac{1}{\sqrt{2\pi\Delta}} \exp\left\{-\frac{(t-t_s)^2}{2\Delta}\right\}$$

Where

$$t_s \propto y_0^{\frac{3}{2}} \exp\left(\frac{x_0^2}{4Dy_0}\right), \Delta \propto y_0^2 \exp\left(\frac{x_0^2}{2Dy_0}\right)$$

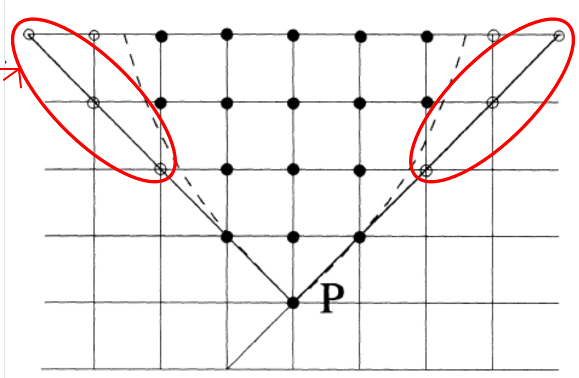
For $x_0 \geq (Dy_0)^{\frac{1}{2}}$ (outside parabolic region)

Search time grows *faster than exponentially*

Δ also increases with x_0

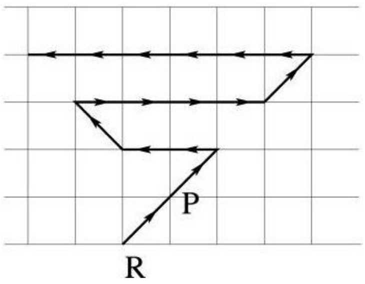
(x_0, y_0) is the initial position of the moth
t_s is the typical search time
Δ is the PDF variance

When search time is large, $(t \gg t_s)$, moth gets "trapped" outside parabolic region

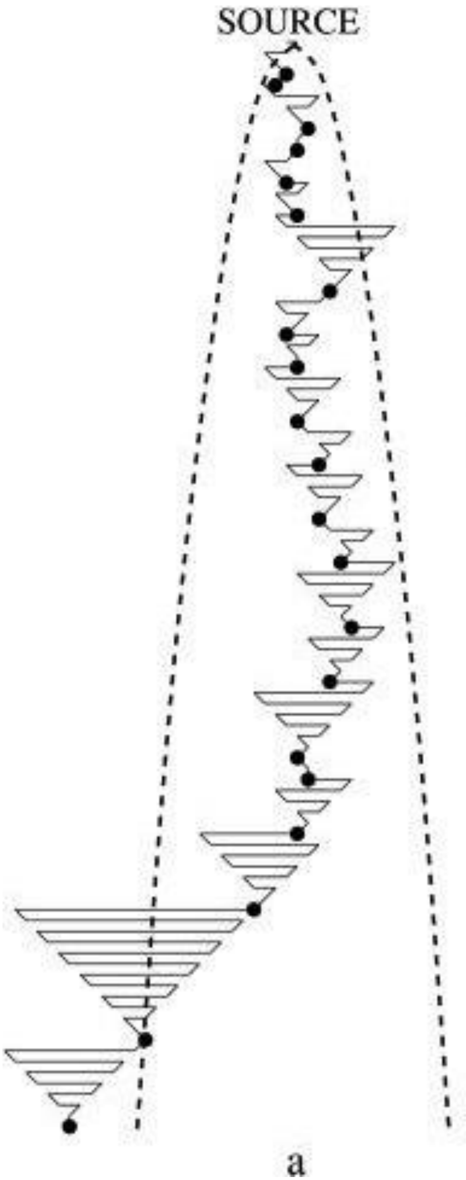


Strategy 2

- Moth moves in casting motion until scent patch is located.
- Moth moves a unit towards direction of located scent and then begins the zigzagging motion until next scent patch is located.



- Cone of possible scent locations collapses as each patch is detected and a movement towards the scent is made.
- Frequency of scent patch detection increases as moth moves closer to source.
- Once detection is frequent enough, moth can take direct path toward source.

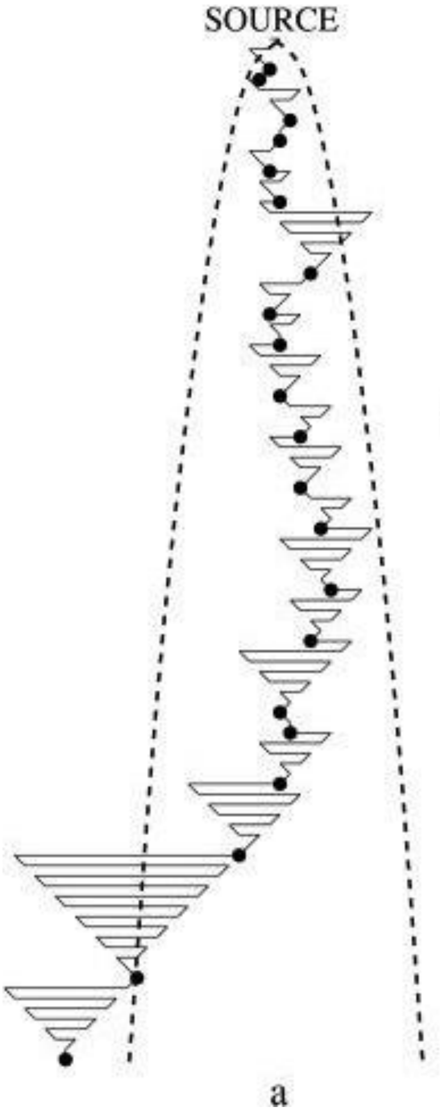


Strategy 2

- The PDF of the "active search" time is:

$$\rho(t) = \frac{1}{4\sqrt{\pi b t}} \exp\left(-\frac{(t-t_s)^2}{2bt}\right)\left(1+\frac{t_s}{t}\right),$$

- Net upwind component of moths velocity is largest immediately after scent detection and decreases with time as $\frac{1}{\sqrt{t}}$.
- Typical search time is: $t_s = a y_0^{5/4}$.
- Typical search time is independent of initial crosswind position.
- With upwind movement, there is a high probability that next patch will be encountered within $|x| \lesssim (Dy)^{1/2}$.

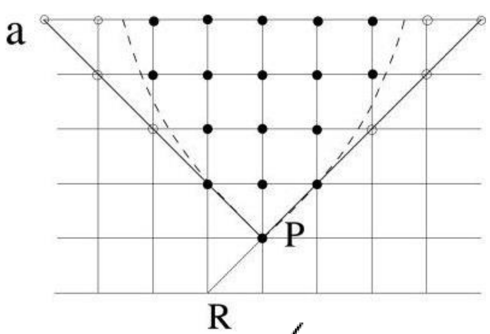


Strategy 3

- Modification of strategy 2.
- PDF for search time is same as strategy 2.

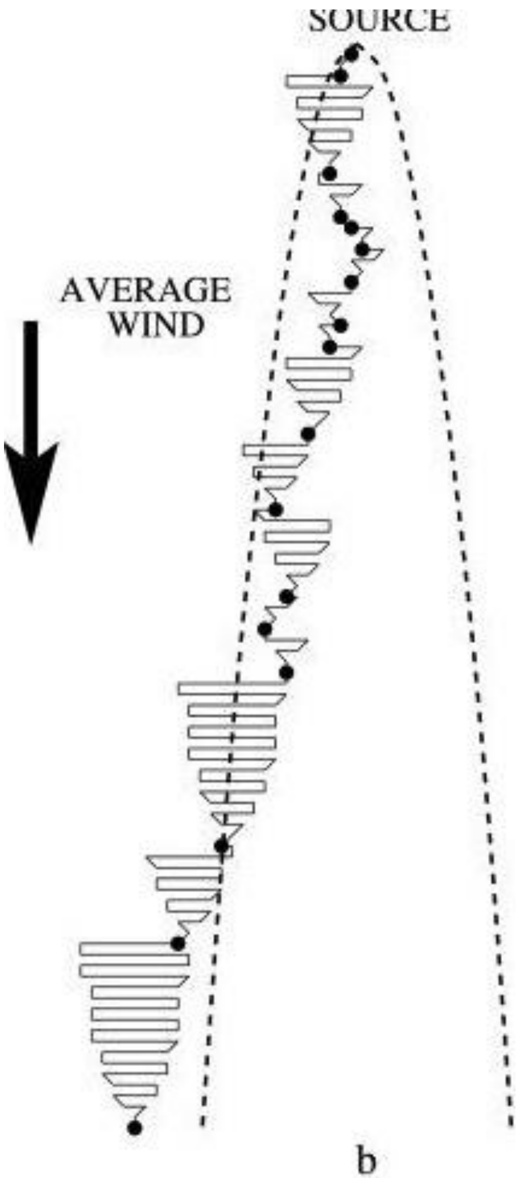
$$\rho(t) = \frac{1}{4\sqrt{\pi b t}} \exp\left(-\frac{(t-t_s)^2}{2bt}\right)\left(1+\frac{t_s}{t}\right),$$

- Difference from strategy 2 is that some search points can be omitted.



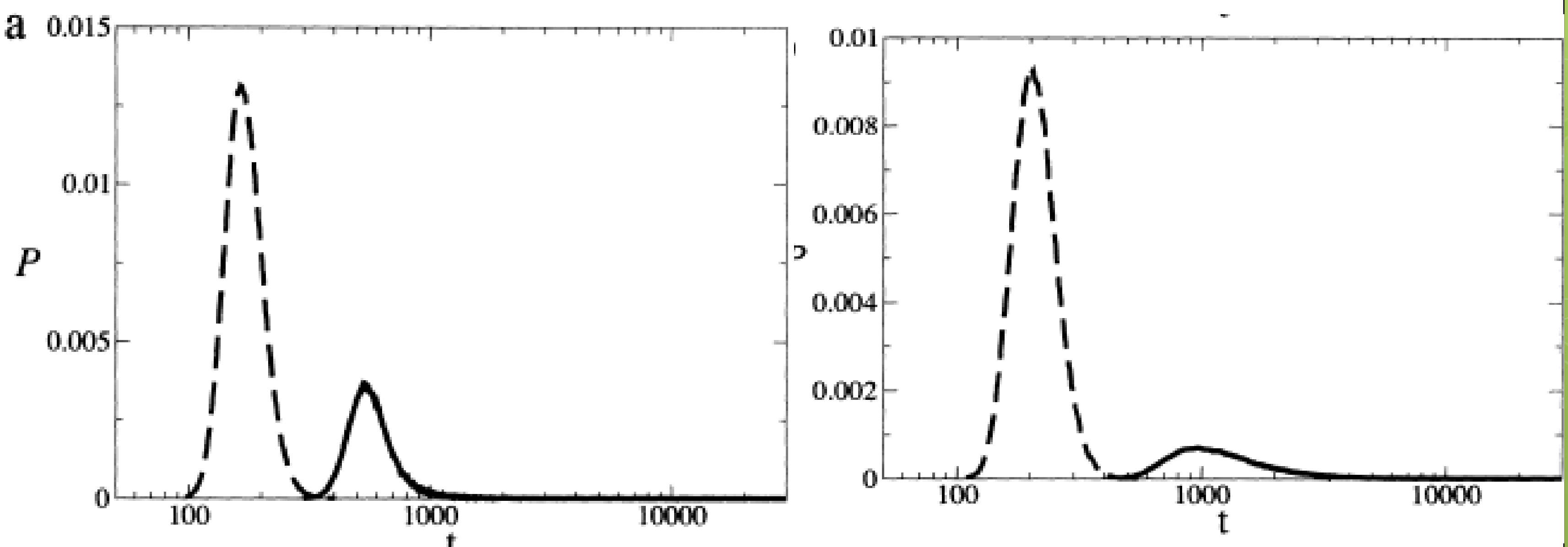
$$(x-x_i)^2 \le 4D(y_i-y)\ln\left(\frac{1}{p_c\sqrt{4\pi D(y_i-y)}}\right),$$

- Typical search time is: $t_s = a_2 y_0^{7/6}$
- Drawback is small possibility to lose the plume.



Results

- Passive search algorithms are inefficient.
- Cross wind motion is essential.



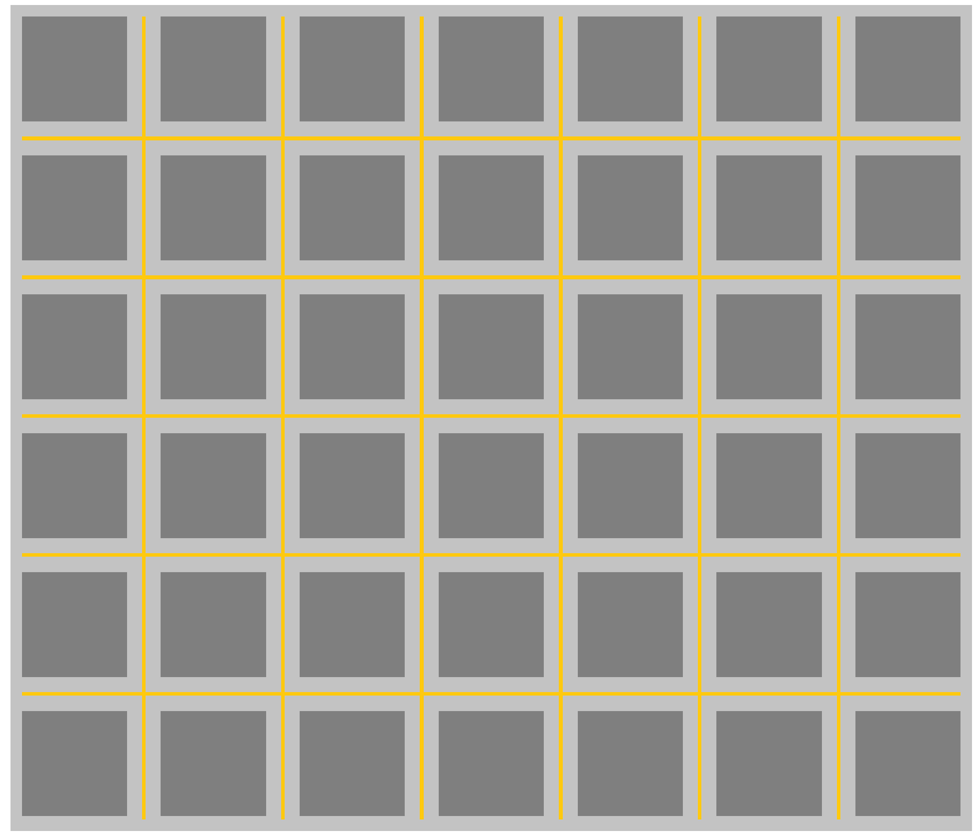
Our Project

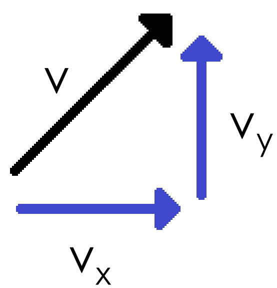
Olfactory Search in Turbulent Flow in a City

- Analysis of how odor distributes given the consistent obstruction by large objects (i.e. buildings).
- Formulation of an analytical model for the distribution of odor patches in a city.
- Collection of data based on a Monte Carlo algorithm for comparison with analytical model.
- Construction of search algorithm for “bomb sniffing” robot.

The Setup

- Assumption that the city is a perfect, square grid.
- Wind Velocity is a given parameter
- Every time interval, an odor patch is released from the source. It's velocity depends on which wind velocity component is affecting it.

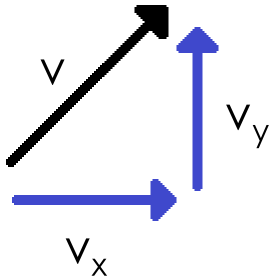




v = Wind Velocity

The Setup

- Assumption that the city is a perfect, square grid.
- Wind Velocity is a given parameter.
- Every time interval, an odor patch is released from the source. It's velocity depends on which wind velocity component is affecting it.
- At each intersection the probability that the smell patch moves in a given direction is weighted by the components of the wind velocity vector.
- The buildings are very tall, and the wind flow at street level is not affected by the turbulent flow above the roof-tops in most cases.
- Given a robot that can detect odor and wind velocity, what is the best search pattern? How can Nature's (the moth's) search pattern be adapted to the environment?



v = Wind Velocity


Y
X

Questions?