ISSN No.: 2454- 2024 (online)

International Journal of Technical Research & Science

Using Sensors Network to Map and Localize Odor Propagation for Airborne Chemical Source

Xiang Gao [*], Acar Levent [**]

Dept. of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO, USA.

Abstract-The article addresses the problem of mapping the continuous particle paths which are most likely taken by the odor particle detected by the concentration sensors and anemometer sensors. The estimated particle paths are useful for odor source localization, which the existing methods for particle path estimation are either at high computational cost, or in discrete form that is incomplete for estimating the localization of the odor source. In this paper, a novel algorithm is proposed that maps the trajectories of chemical particle on the fluid flow and smooth the particle paths based on the primary paths using additional sensors.

Key words: odor path, odor source localization, online mapping, array sensors

1. INTRODUCTION

Odor localization is performed by many type of mechanism, in a variety of methods. Sensing and responding to chemical is now a burgeoning area of robotics research. These applications have been tackled by researchers in a variety of methods. Most of their works have focused on mobile robots to detect plume and trace plume [1and2]. Odor dispersal occurs through carriage by the fluid current. Many researchers has used a chemical gradient method that is acquiring the plume firstly, and then moving upwind along the plume [3 and 4]. But, their limitation are that robots must follow the plume along its entire length, which is time consuming and may not be possible in some environment where a well-defined plume does not be formed. In addition, the effects of objects and walls on both airflow and robot mobile are often neglected [5].

In this paper, we have attempted to overcome these limitations. By exploiting more information from the environment in addition to local wind and chemical reading, the airflow map and chemical distribution can be derived first, then basing the aforementioned result a 'sense-map-plan-act' control scheme is built [6].

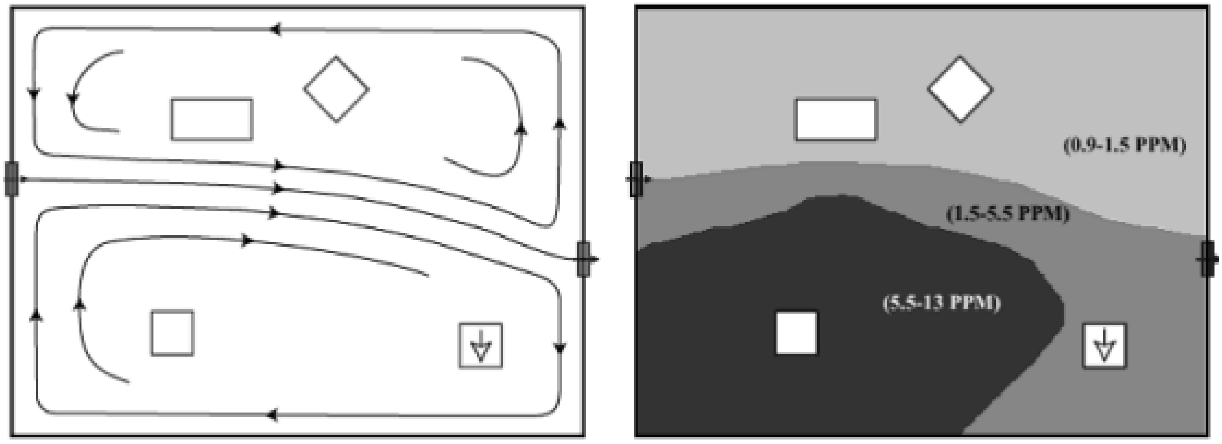

Fig. 1.1 Airflow and Odor Distribution in a Simple Environment

However, when the sensor system collects more data, the problem of updated mapping is faced to be solved. Because the information is local, which does not affect the whole mapping, we just need update the airflow and chemical dispersion around the sensors that gather new data. Neural networks using splines function can update its weights to satisfy the new data from sensors.

2. THE METHODOLOGY OF PLOTTING AND STOPPING ROLES

Step 1: Confirm the Order of the Sensors

There are several sensors in the room where we want to detect the source, we should got the correct order from the all permutations. For excluding the incorrect and contradictory order, we have set the rule that we think the air flow can propagate along the shorter route. The process is showed below:

- Using the cubic parameter interpolation method, we can get the curve between any two sensors.
- Calculate the length of the above-mentioned curve, such as L_{AB}.
- We can get the length between the two sensors, when the sensors' order is flipped, such as L_{BA}.
- Compare the lengths when the orders of the sensors are different. We think the shorter one is correct order. Such as If $L_{AB} < L_{BA}$, the correct order is A to B.
- Do a loop to get the correct order between each two sensors.

Step 2: Interpolate the Paths that Connect the Sensors

In our modeling, we assume that there're only very few sensor data available, since it is difficult to place sensors and collect data in locations that aren't very accessible and user friendly. With a few sensors, we collect the concentrations levels of the chemical particle, the wind direction, and the local dissipation of the particle. Using cubic Hermite parameter interpolation, we can get the paths that connect the sensors. Moreover, we can adjust the parameter to get the extrapolation.

The problem with Hermite interpolation is the need to specify the derivatives at the endpoints of each section of the curve. Suppose the curve has n+1 data points (x0,y0), ⋯， (xn, yn), and we wish to parameterize the cubic to allow complex features. Then we must specify x'(ti) and y'(ti) for each i= 0, 1, ⋯,n , where (xi, yi)= (x(ti) , y(ti)). This is not as difficult as it would first appear, however, since each portion can be generated independently, provided that we ensure that the derivatives at the endpoints of each portion match those in the adjacent portion. Essentially, then, we can simplify the process to one of determining a pair of cubic Hermite polynomials in the parameter t, where t0=0 and t1=1, given the endpoint data (x(0), x(1), y(0), y(1)) and the derivatives dy/dx(at t= 0) and dy/dx(at t= 1). The natural form for determining x(t) and y(t) required we specify x'(0), x'(1), y'(0) and y'(1). The explicit Hermite curve in x and y required specifying only the quotients

$$\frac{dy}{dx}(t=0)=\frac{x'(0)}{y'(0)} \quad \text{and} \quad \frac{dy}{dx}(t=1)=\frac{x'(1)}{y'(1)}$$

By multiplying x'(0) and y'(0) by a common scaling factor, the tangent line to the curve at (x(0), y(0)) remains the same , but the shape of the curve varies.

To further simplify the process, the derivative at an endpoint is specified graphically by describing a second point, called a guide point, on the desired tangent line. The farther the guide point is from the node, the more closely the curve approximates the tangent line near the node.

The node occur at (x0, y0) and (x1, y1), the guide point for (x0, y0) is (x0+α_0, y0+β_0), and the guidepoint for (x1, y1) is (x1-α_1, y1-β_1). The cubic Hermite polynomial x(t) on [t0, t1] must satisfy

$$X(0)= x0,\ x(1)= x1,\ x'(0)= \alpha_0,\ x'(1)= \alpha_1$$

It is easily verified that the unique cubic polynomial satisfying these conditions is

$$x(t)=[2(x_0-x_1)+(\alpha_0+\alpha_1)]t^3+[3(x_1-x_0)-(\alpha_1+2\alpha_0)]t^2+\alpha_0 t+x_0.$$

In the similar manner, the unique cubic polynomial for y is

$$y(t)=[2(y_0-y_1)+(\beta_0+\beta_1)]t^3+[3(y_1-y_0)-(\beta_1+2\beta_0)]t^2+\beta_0 t+y_0 \qquad 2.1$$

Popular graphics programs use this type of system for their freehand graphic representations but in a slightly modified form. The Hermite cubic are described as Bezier polynomial, which incorporate a scaling factor of 3 when computing the derivatives at the endpoints. This modifies the parametric equations to

$$x(t)=[2(x_0-x_1)+3(\alpha_0+\alpha_1)]t^3+[3(x_1-x_0)-3(\alpha_1+2\alpha_0)]t^2+3\alpha_0 t+x_0$$

In the similar manner, the unique cubic polynomial for y is

$$y(t)=[2(y_0-y_1)+3(\beta_0+\beta_1)]t^3+[3(y_1-y_0)-3(\beta_1+2\beta_0)]t^2+3\beta_0 t+y_0 \qquad 2.2$$

We can construct a set of cubic Bezier curves ($C_0, C_1, \cdots, C_{n-1}$) based on the parametric equations in（2.2）.Where C_i is represented by $x_i(t), y_i(t)=(a_0^i+a_1^i t+a_2^i t^2+a_3^i t^3, b_0^i+b_1^i t+b_2^i t^2+b_3^i t^3)$

Step 3: Some Rules to Remove Some Unreasonable Points and Their Links.

Assuming the each two points can have a link, we can get countless new points and links. So, we should set up some rules to remove some unreasonable points and their links.

Check Order: All the links should obey the region order. In our case, all links should follow the order (region 1 to region 3, region 1 to region 2 to region 3).

Check Process: We can get only one link derived from one link. It's not allowed that there are two different links from one point.

Check Association: Firstly, we give the definition of point association. If one point is generated by another point along another one's perpendicular line, we think there are belonged to a same group. We think there are not link between each two points in the same group.

Check Group: If the link with the other existed one have same endpoints and beginning points that are belonged to the same group. We think the new link should be removed.

Check Original Points (Sensors): Every muti-links should go through one point that is the location of any one sensor.

We should check each above rule. If one of rules isn't obeyed, the link should be removed. Finally, we find if there are n sensors, we will have n*(n-1) links and n multi-links.

Step 4: Scaled Correctly Points on the Path Showing the Speed of Propagation

In the points where there are the sensors, we can get the speed of the airflow from the sensor data. But, we want to show the speed of every point in the whole room in our map.

In the step 2, we have got the paths on which there are a few sensors. In this step, we want to scale correctly points on the path showing the speed of propagation. Using the speed of the points where the sensors are located, we can get the speed of propagation on every point. In our case, we think the speed has the linear change along the propagation paths. The beginning point is the sensor point as we know the speed in these points. We draw the speeding points using the equal time interval. Through the speed value and the time interval, it is simple to get the next speed point. Step by step, we can finish the all speed point along the propagation direction on the path. In the map, we can find the longer the distance between to speeding point indicate the speeding on this part is faster than the other one.

Step 5: Map the Whole Room Using Interpolation and Extrapolation

In the step 2, we have got the paths on which there are a few sensors. But, we want to topographically generate the flow diagram on a map of the surroundings and determine the probable source location of the chemical particles. There are two different cases when we map the whole room. One is the region between two paths; the other is the region between a path and the boundary. When we meet the first case, we can use the method of bisection. Using two points from two different paths we can get the midpoint. We apply this method in a loop; we can compute the whole path. As far as the other region, we use a different method that based on the former method. We calculate the normal vector of each point with a path that is belonged to the former region in order to get the crossing point. Then, we generate a new point in the opposite direction. The distance from initial point and crossing point is proportional with the distance from initial point to the new point.

We're able to improve on the accuracy of the route as well as generate more viable routes. With the added accuracy and flexibility, we're also able to eliminate unfeasible options and conflicting sources.

Because every path is consisted of many discrete points, we can use dichotomization to get several new points. Then, the new points together form a new path. Finally, we can map the whole room in order to mark the chemical situation everywhere.

Step 6: Considering Boundary Condition to Modify and Stop the Airflow Path

Several difficulties arise when formulating a shape transformation. The instance in which boundary information signals the presence of a shape is not clear. The decision of which boundary information to include and exclude in a description of shape is also nontrivial. This decision process poses both a local and a global problem. Over a localized region of a shape, the transformation should separately shape boundary information elements related to different shapes, even if shapes extend over a very wide spatial region. On a more global scale the transformation must be able to integrate local shape encodings as subparts of the same global structure.

We know the boundary of room can affect the airflow's propagation. If there is not the boundary, the path would pass it. But, in fact, the propagating paths are affected by the boundary, they will spread along the boundary. The airflow near the boundary cannot swerve suddenly, so we can assume an area that is very near the boundary, if the airflow enters into the area, it will be changed by the boundary. If the airflow is out of this area, it will not be affect by the boundary.

I will introduce a method using proportional squeeze. If there is not the boundary existed, the airflow will go beyond it. We find using *tanh* function to squeeze the distance between boundary and airflow can make the airflow approximate the real airflow path. Firstly, we calculate the normal vector of each points that are beyond the inside boundary, and the normal will have a crossing point with the inside boundary. Secondly, we can calculate the distance between the point and the crossing point, then we make the distance shorter using tanh function . Finally, we can get a new point that is in the boundary area . We find the tanh can make our path satisfy this really condition that the closer the point with the inside boundary, it will get more effect by the boundary. When the point is near the inside boundary, the path just have a tiny change with the path without considering the boundary. When the points is far from the boundary, the path will be squeezed sharply. We apply this method in a loop, we can compute the whole path. When the airflow curve is in the boundary region, I stop the airflow curve when it is parallel with the boundary.

3. SMOOTH THE PARTICLE PATHS USING MORE SENSORS

In the cases that more sensors are scattered into the considered area to get more precise particle paths, we should propose the algorithm to use the additional sensors to smooth the particle paths. We get more data in some new positions $[(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)]$, where the sensors can measure the wind velocity and direction $U_i(x_i, y_i) = [U_{xi}(x_i, y_i), U_{yi}(x_i, y_i)]$, where $U_{xi}(x_i, y_i)$ is the velocity component on X coordination, $U_{yi}(x_i, y_i)$

is the velocity component on Y coordination, odor concentration $[C(x_1, y_1), C(x_2, y_2), \ldots, C(x_n, y_n)]$ and the concentration gradient $\delta(x_i, y_i) = [\delta_{\perp}(x_i, y_i), \delta_{\parallel}(x_i, y_i)]$, where $\delta_{\parallel}(x_i, y_i)$ is the component on the fluid flow's direction and $\delta_{\perp}(x_i, y_i)$ is the component on the perpendicular direction . Obviously, the primary paths function gotten in the chapter 2 can't satisfy the data from the new positions [7]. The algorithm need modify the primary paths' parameters to fit the new constraints. We use a simple example to demonstrate the algorithm as below.

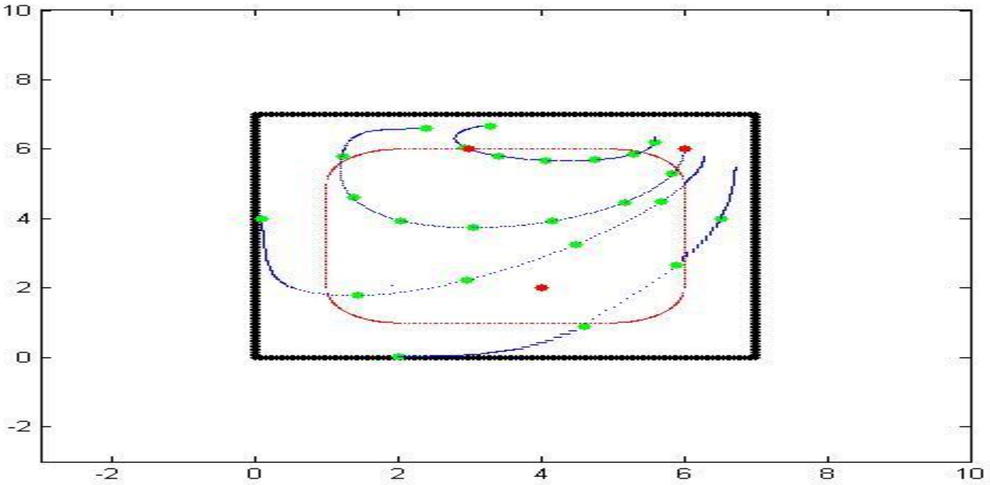

Fig. 3.1 Primary Air-Borne Particle Paths Going Through Two Sensors and an Additional Sensor

In the fig. 3.1 the bottom sensor is a new sensor that is fixed after the primary particle paths are gotten. The propagation between two primary sensors can be remodeled to two piecewise particle paths via two primary sensors and an additional sensor, then we use our novel algorithm update the parameters of primary paths to close the piecewise particle paths. In brief, the object is to minimize the error,

$$\begin{aligned} x(t) &= (2(x(0) - x(\mathrm{n})) + 3(\delta x(0) + \delta x(\mathrm{n})))t^3 + 3(x(\mathrm{n}) - x(0)) - 3(\delta x(\mathrm{n}) + 2\delta x(0)))t^2 + 3\delta x(0)t + x(0), \\ y(t) &= (2(y(0) - y(\mathrm{n})) + 3(\delta y(0) + \delta y(\mathrm{n})))t^3 + 3(y(\mathrm{n}) - y(0)) - 3(\delta y(\mathrm{n}) + 2\delta y(0)))t^2 + 3\delta y(0)t + y(0). \end{aligned} \quad 3.1$$

where $x(0) \le x(t) \le x(\mathrm{n}), y(0) \le y(t) \le y(\mathrm{n})$, this piecewise path is denoted by L_1 .

$$\begin{aligned} x(t) &= (2(x(\mathrm{n}) - x(1)) + 3(\delta x(\mathrm{n}) + \delta x(1)))t^3 + 3(x(1) - x(\mathrm{n})) - 3(\delta x(1) + 2\delta x(\mathrm{n})))t^2 + 3\delta x(\mathrm{n})t + x(\mathrm{n}), \\ y(t) &= (2(y(\mathrm{n}) - y(1)) + 3(\delta y(\mathrm{n}) + \delta y(1)))t^3 + 3(y(1) - y(\mathrm{n})) - 3(\delta y(1) + 2\delta y(\mathrm{n})))t^2 + 3\delta y(\mathrm{n})t + y(\mathrm{n}). \end{aligned} \quad 3.2$$

where $x(\mathrm{n}) \le x(t) \le x(1), y(\mathrm{n}) \le y(t) \le y(1)$, this piecewise path is denoted by L_2.

In common, the error term should be $E = \frac{1}{2}\sqrt{(\sum(\int_{L_1} y \cdot x'\mathrm{dt} + \cdots + \int_{L_n} y \cdot x'\mathrm{dt}) - \int_L y \cdot x'\mathrm{dt})^2}$, when there are n additional sensors set between the primary paths.

Depending on the expression of error term, the minimization of the error can be gotten from

$$\begin{aligned} \int_L y \cdot x'\mathrm{dt} &= \int_0^1 ((2(y(0) - y(1)) + 3(\delta y(0) + \delta y(1)))t^3 + (3(y(1) - y(0)) - 3(\delta y(1) + 2\delta y(0)))t^2 + \\ &3\delta y(0)t + y(0))(2(x(0) - x(1)) + 3(\delta x(0) + \delta x(1)))t^3 + 3(x(1) - x(0)) - 3(\delta x(1) + 2\delta x(0)))t^2 + \\ &3\delta x(0)t + x(0))' dt \\ &= k_1 x(1) y(1) + k_2 x(1) + k_3 y(1) + k_4 \end{aligned}$$

,where k_1, k_2, k_3, k_4 are determined by the data from sensors.

We use $\Delta x(1), \Delta y(1)$ to denote the updated parameters. Since $\Delta x(1), \Delta y(1)$ is on the normal line of the sensor, then $\Delta x(1), \Delta y(1)$ should satisfy the equation:

$$y = \frac{\delta y(1) - y(1)}{\delta x(1) - x(1)}(x - x(1)) + y(1) \quad 3.3$$

Use $\Delta x(1), \Delta y(1)$ to substitute $x(1)$, $y(1)$ in Eq. 3.1, we can get the updated particle path,

$$\begin{aligned} x(t) &= (2(x(0) - \Delta x(1)) + 3(\delta x(0) + \delta x(1)))t^3 + 3(\Delta x(1) - x(0)) - 3(\delta x(1) + 2\delta x(0)))t^2 + 3\delta x(0)t + x(0), \\ y(t) &= (2(y(0) - \Delta y(1)) + 3(\delta y(0) + \delta y(1)))t^3 + 3(\Delta y(1) - y(0)) - 3(\delta y(1) + 2\delta y(0)))t^2 + 3\delta y(0)t + y(0). \end{aligned} \quad 3.4$$

In this example, using the above approach, we can update $x(1)$, $y(1)$ to $\Delta x(1), \Delta y(1)$. Through the Eq. 3.4 we can get the new particle paths that showed in the red line in figure 3.

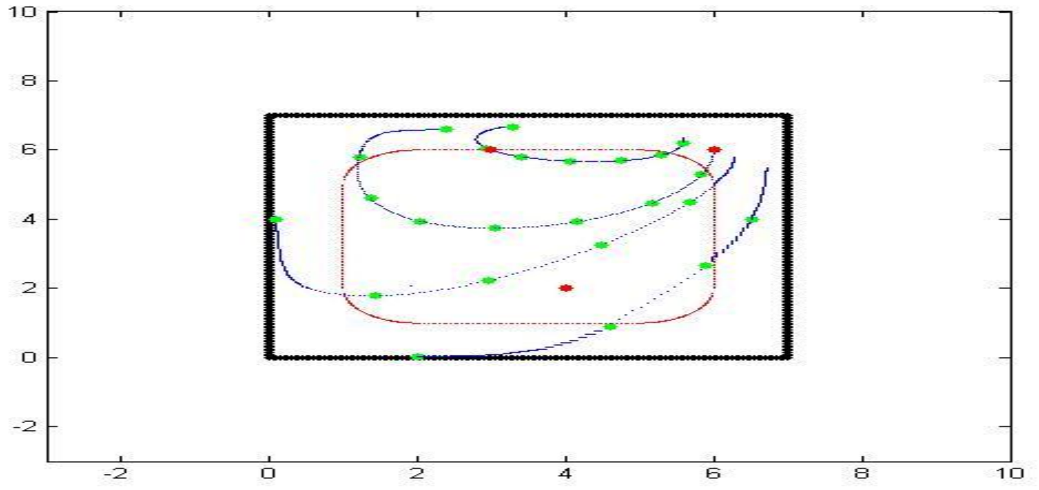

Fig. 3.2 Updated Air-Borne Particle Paths Going Through Two Sensors and an Additional Sensor

4. EXPERIMENTAL EVALUATION

The main purpose of this chapter is to present the experimental results in order to test and verify our developed method. From the Google Earth, a real map of Missouri University of science and technology can be got. Using the edge detection technology, the processed map only exits the main buildings that are considered obstacles to effect the chemical particle propagation. The unimportant pixels can be erased. Figure 4 shows the result of real map processed by edge detection approach. We assumption the wind velocity is 10m/s from southwest.

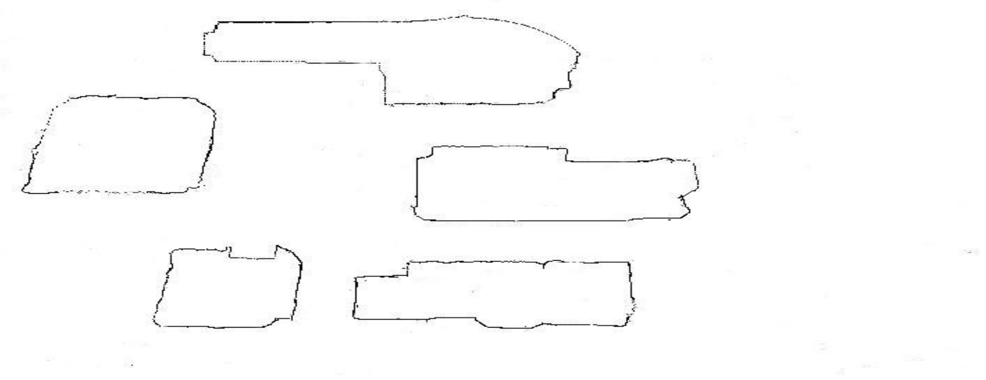

Fig. 4.1 A Real Map of Missouri University of Science and Technology Processed by Edge Detection Method

In our experiment, we first scatter some sensors in the considered area, then through interpolation and extrapolation method the primary particle paths are get based on the data from the two sensors. When the other more sensors are scattered in to this area, the particle paths should be modified based more data from the new sensors. At last, we use COMSOL to simulation the chemical particle's propagation in the same conditions. Comparing the results from COMSOL to results of particle paths maps from different numbers of sensors, we find in the considered area the more sensors we used the more precise result we get.

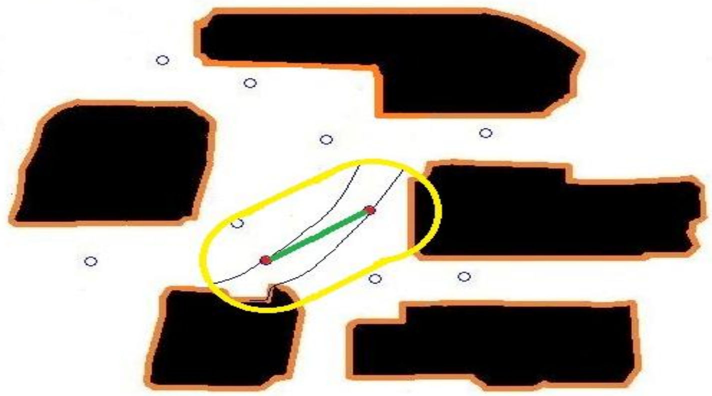

Fig. 4.2 The map of particle paths using two sensors

In the fig. 4.2, we place two sensors in the considered area to map the particle paths. In the figure 6, we scattered two more sensors in this area, then use our proposed parameter updated rules to map the new particle paths. Comparing the figure 7 the real particle paths, we can include that the updated rules are effective. In the other words, the accuracy of the particle paths is advanced by the more sensors.

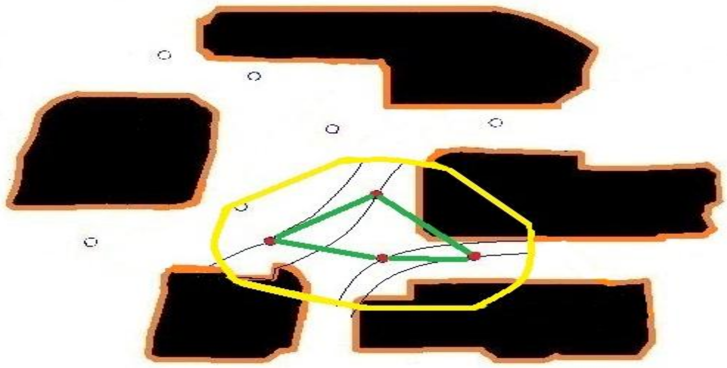

Fig. 4.3 The map of particle paths using four sensors

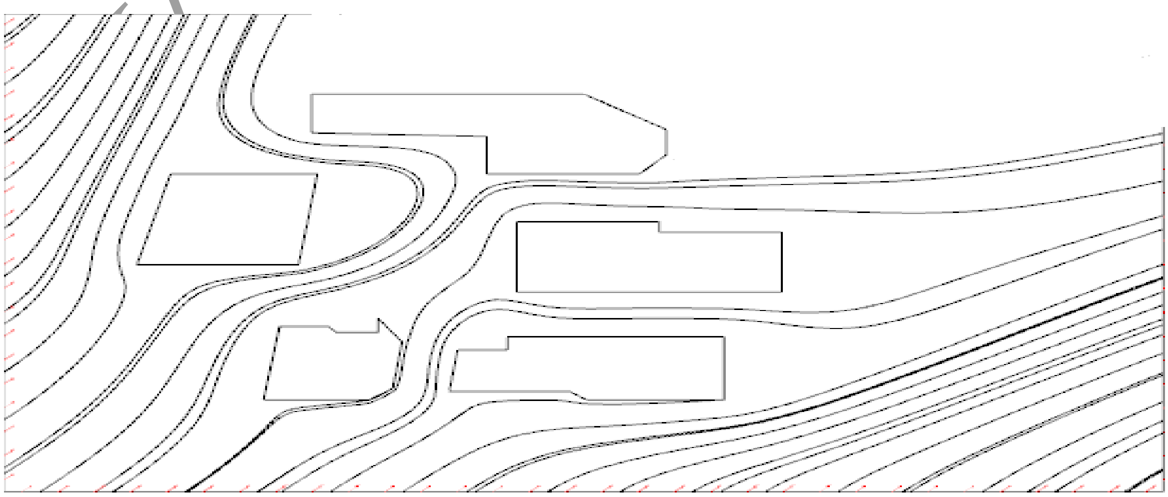

Fig. 4.4 Air-Borne Particle Paths Going Through Ten Sensors in a Real Map Processed by COMSOL

CONCLUSION

There are many useful and humanitarian applications that can locate the source of a chemical source. Currently, the majority of work in this area uses reactive control schemes that track an odor plume along its entire length, which is slow and difficult in cluttered environments. This paper employs a high-level control scheme. The interpolation and extrapolation method is used to model the particle path in the sensors' environment. Then a reasoning system use the path model to get the velocity, chemical concentration at any point on the map and predict the most probable locations of the odor source. This approach has been shown to be effective for odor localization in a known environment, without the need for the robot to travel to the source.

With the further development there is great potential for this approach to lead to many valuable applications by generalization to a wider range of environmental configurations. The paper present development to solve the problem there exist obstacles and opening in the environment. The approach gives the mode of the particle path surrounding the obstacles and openings. The development has successfully applied in general environment, because the propagation of the chemical particle can go through obstacles and opening in actual case. In addition, this paper is the first example of using interpolation and extrapolation method to model the particle path that applied in a real environment.

In addition, the result of simulating the particle path in the real map of campus is approximate with the result from the fluid dynamic analysis software, COMSOL. Hence, this approach can approximately map the particle path in real cases. Future work will concentrate on two areas of development: development of more general particle path modeling algorithm, and implementation of a range data acquisition system to enable autonomous map the building.

REFERENCES

[1] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[3] E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," IEEE Trans. Antennas Propagat., to be published.

[4] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style)," IEEE Transl. J. Magn.Jpn., vol. 2, Aug. 1987, pp. 740–741 [Dig. 9th Annu. Conf. Magnetics Japan, 1982, p. 301].

[5] J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility (Periodical style)," IEEE Trans. Electron Devices, vol. ED-11, pp. 34–39, Jan. 1959.

[6] X. Gao, L. Acar, Detection and Tracking of Odor source in Sensor Networks Using Reasoning System, Journal of Automation and Control Engineering (JOACE, ISSN:2301-3702).

[7] X. Gao, L. Acar, Multi-sensor Integration to map odor distribution for detecting chemical source, Sensors, July, 2016.