Received March 2004
Revised November 2004
Accepted November 2004

Optimal placement of sensors for contaminant detection based on detailed 3D CFD simulations

Rainald Löhner and Fernando Camelli
School of Computational Science and Informatics, George Mason University, Fairfax, Virginia, USA

Abstract

Purpose – Develop a method for the optimal placement of sensors in order to detect the largest number of contaminant release scenarios with the minimum amount of sensors.

Design/methodology/approach – The method considers the general sensor placement problem. Assuming a given number of sensors, every release scenario leads to a sensor input. The data recorded from all the possible release scenarios at all possible sensor locations allow the identification of the best or optimal sensor locations. Clearly, if only one sensor is to be placed, it should be at the location that recorded the highest number of releases. This argument can be used recursively by removing from further consideration all releases already recorded by sensors previously placed.

Findings – The method developed works well. Examples showing the effect of different wind conditions and release locations demonstrate the effectiveness of the procedure.

Practical implications – The method can be used to design sensor systems for cities, subway stations, stadiums, concert halls, high value residential areas, etc.

Originality/value – The method is general, and can be used with other physics-based models (puff, mass-conservation, RANS, etc.). The investigation also shows that first-principles CFD models have matured sufficiently to be run in a timely manner on PCs, opening the way to optimization based on detailed physics.

Keywords Optimization techniques, Sensors, Hazardous materials, Mathematical analysis

Paper type Research paper

1. Introduction

The intentional or unintentional release of hazardous materials can lead to devastating consequences. In order to mitigate the degree of damage it is necessary to detect the presence of hazardous materials.

Two different types of sensors are commonly used in detecting hazardous materials: point sensors and standoff sensors. The first class of sensors is characterized by high sensitivity, accurate identification of specific agents and low cost per unit. However, these sensors are very expensive to maintain. The second class includes standoff sensors (or non-invasive sensors) and is usually based on laser beam technology. These sensors can cover larger areas, but they are very expensive and do not accurately identify specific agents.

The number of sensors is typically limited by several constraints, notably budget and availability of locations for placement. It is therefore of great importance to deploy the limited number of sensors in such a way that they are able to detect the maximum

This work was partially supported by DTRA.

Engineering Computations: International Journal for Computer-Aided Engineering and Software
Vol. 22 No. 3, 2005
pp. 260-273

0264-4401
DOI 10.1108/02644400510588076

possible number of release scenarios. This question has received increased attention in recent years. Various optimization techniques have been proposed, ranging from Monte-Carlo simulations with Gaussian plume models (Garten *et al.*, 2003), genetic algorithms with data compression models of 3D simulations (Obenshain *et al.*, 2004) and two-point space correlations of Gaussian plume models (Sykes and Chowdhury, 2004). Most of these methods require thousands of simulations, restricting them to rather simple physics that can be incorporated into a fast running model.

This short note is intended to illustrate a simple method that can provide a solution to the sensor placement problem with the minimum possible number of simulations, making it feasible to incorporate first-principle, 3D unsteady flow/contaminant transport simulations.

2. Optimal sensor placement

Consider the general sensor placement problem. If we assume a given number of sensors, every release scenario (location and amount of release, current meteorological conditions, etc.) will lead to a sensor input. The data recorded from all the possible release scenarios at all possible sensor locations allows the identification of the best or optimal sensor locations. Clearly, if only one sensor is to be placed, it should be at the location that recorded the highest number of releases.

This argument can be used recursively by removing from further consideration all releases already recorded by sensors previously placed. The procedure has been sketched in Figure 1. The case shown has three possible release locations, three possible wind directions and three possible sensor locations. For every release scenario (location and wind direction), a run is performed and the data recorded at all sensors is stored. Once all these data have been gathered, the sensor that was able to detect the

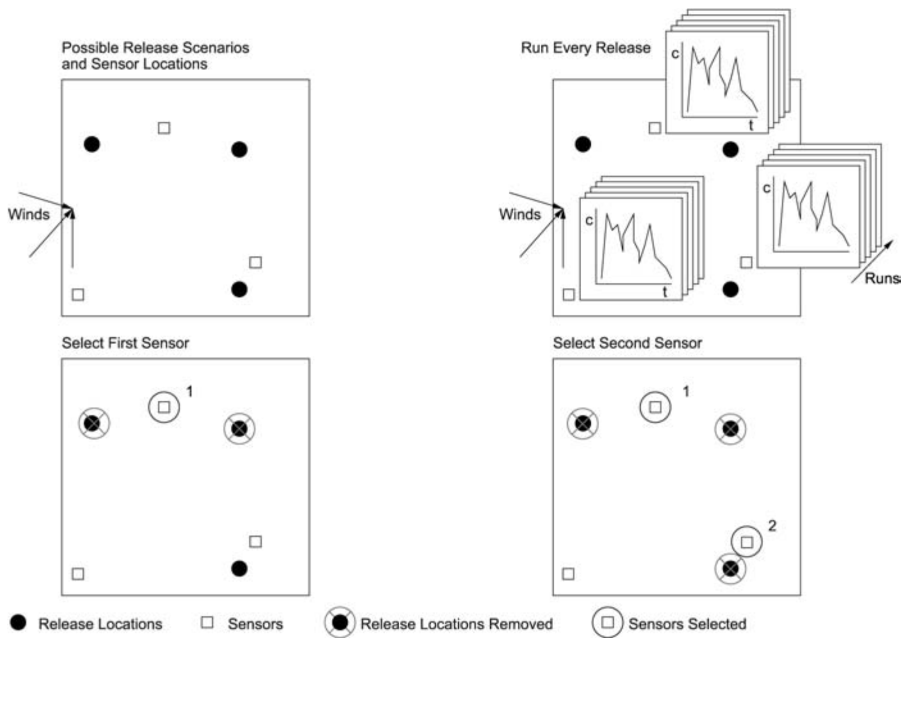

Figure 1.
Optimal sensor placement

highest amount of cases is selected (number 1 in Figure 1). All the release scenarios sensed by this sensor are removed from further consideration. The number of release scenarios sensed by the remaining sensors is again compared, and the sensor that was able to detect the highest amount of cases is selected next (number 2 in Figure 1). The procedure is repeated recursively until no undetected release cases are left, or the available sensors have been exhausted. A pseudo-code summary of the procedure may be found in the Appendix.

3. Calculation of sensor input

The input parameters for possible release scenarios are manifold: meteorological conditions (notably wind direction), type and amount of contaminant, and release location (Hanna *et al.*, 1982; Stern *et al.*, 1984; Arya, 1998, 1999). As with any other disciplines, a number of models with different degrees of approximation and fidelity can be employed for the prediction of contaminant transport. In ascending order of fidelity, we mention puff models (Hanna, 1998; Witlox, 1999; Hanna *et al.*, 2003), mass-conservation models (Sherman, 1978; Prabha and Mursch-Radlgruber, 1999), Reynolds-averaged 3D CFD models (Coirier and Jorgenson, 1996; Kastner-Klein *et al.*, 2000; Lim *et al.*, 2000), and detailed (very large eddy simulation) 3D CFD models (Hanna *et al.*, 2002; Camelli *et al.*, 2003, 2004; Obenshain *et al.*, 2004). In the present case, we have chosen to use the last option. We remark again that for each of these (number of releases × number of wind directions) scenarios, a separate computational fluid dynamics (CFD) contaminant transport run is performed.

Recent advances in CFD codes and hardware have made it possible to perform large-scale, high-resolution 3D runs in a reasonable amount of time (Löhner, 2001; Camelli *et al.*, 2003, 2004; Löhner *et al.*, 2003). Notable advances in CFD codes include the use of explicit multistage advection within projection-type solvers (Löhner *et al.*, 2003; Löhner, 2004), which reduces flow solver CPU requirements by a factor of 1 : 5, and the use of dynamic deactivation (Löhner and Camelli, 2004), which yields speed-up factors of 1 : 5-1 : 20 for transport simulation. Notable advances in hardware have been PCs with clockrates exceeding 3.0 GHz and RAM exceeding 2.0 GB.

The flowfields for different meteorological conditions (wind direction, atmospheric conditions, humidity, temperature of building walls, emissions of heat exchangers, etc.) are typically calculated and stored on supercomputers. Given that in some cases up to an hour of real time needs to be computed, these runs can consume a considerable amount of computer time. The effects of different release scenarios (type, amount, location, etc.) are then simulated for these pre-stored flowfields. Algorithmic improvements (Löhner *et al.*, 2003) have enabled high fidelity CFD calculations of each one of the release scenarios in a matter of minutes on PC platforms, even for grids in excess of a million elements. This implies that on networks of PCs, or PC clusters, hundreds of release scenarios can be simulated per day.

Sensors are simulated in CFD codes by recording the time history of the concentration at the location of the sensors. Then, for each of the different release scenarios, the time history of all possible sensors in the field is recorded and stored. A sensor is assumed to activate (or "go hot" as it called) once the concentration exceeds a threshold level.

4. Selection of release scenarios

The choice of sensors will be heavily influenced by the release scenarios chosen. Release scenarios are given by a combination of:

Release

- Type (material, amount,...);
- Location(s);

Meteorological

- Wind (velocity, direction,...);
- Temperature;
- Humidity, rain,...;
- Surface heating/absorption;

For most applications, the worst-case scenario is given by a stable atmosphere with light and variable winds. It is in these conditions that contaminant clouds tend to linger the longest at low altitudes, dissipating only slowly. Given that the number of release scenarios can quickly reach very large numbers (number of wind directions $\times$ number of release locations $\times$ number of material types/amounts released $\times \ldots$), and that each one of them requires a 3D CFD simulation, a judicious choice of the most likely (and most harmful) scenarios has to be chosen. As far as wind direction is concerned, at many geographical locations winds tend to have a few predominant directions. As far as release location is concerned, the maximum harm will typically occur if the release location happens to be upstream. In this way, the number of possible release scenarios can be reduced measurably.

5. Examples

The procedure outlined above was applied to several examples. In each of these, the flow physics were described by the incompressible Navier-Stokes Equations with the Smagorinsky turbulence model. A logarithmic profile was applied as inflow boundary condition with a mean velocity of 2 m/s at a height of 10 m. The finite element code FEFLO, a general-purpose flow solver code, was used for both the flow field and dispersion calculations. This particular code has been repeatedly benchmarked and compared to experiments (Camelli and Löhner, 2000; Hanna *et al.*, 2002; Camelli *et al.*, 2003, 2004) and is routinely used for production runs. However, we emphasize that any other CFD tool capable of accurate dispersion calculations could have been used instead. The flow field was pre-calculated and stored in order to speed up the time for each of the dispersion runs. The runs were performed on PCs with Intel P4 chips running at 3.20 GHz, and with 1 GB RAM, Linux OS and Intel compiler.

5.1 Residential blocks

Consider the intentional release in an area representative of an inner city composed of three by two blocks. The geometry definition and the surface mesh are shown in Figure 2. The mesh consisted of approximately 0.85 Mtet. The release time was assumed to be 60 s, and the simulations were carried out for at least 600 s of real time. Each of the dispersion simulation runs took between 4 and 20 min. A typical result is shown in Figure 3. A total of 65 sensors, shown in Figure 4, were placed at possible locations. Twenty dispersion locations were defined, depending on the predominant wind direction. Figure 5 shows the release locations for wind coming from the south/east, and for wind coming from the west. Depending on the number of wind direction/release scenarios considered and the sensor threshold, different optimal sensor numbers and placements are obtained. Figure 6 shows the optimal sensor placement for purely south easterly wind: note that only seven optimal sensors are required to sense all release scenarios with a measurement threshold of $c_t = 0.002$,

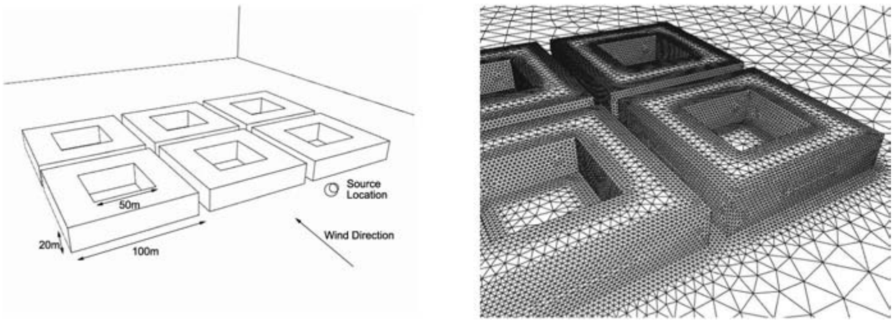

Figure 2.
Problem definition and detail of surface mesh

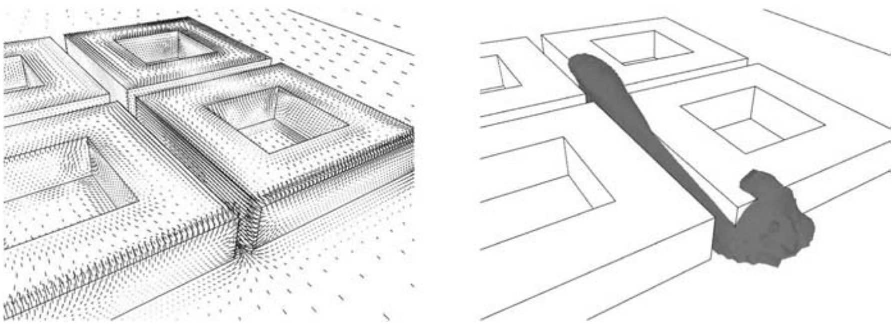

Figure 3.
Surface velocities and typical iso-surface of concentration

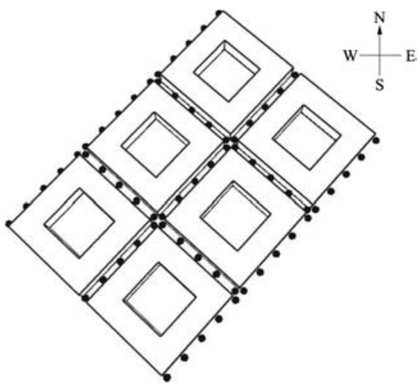

Figure 4.
Sensor placement

whereas 12 sensors are required for a measurement threshold of $c_t = 0.020$. Furthermore, the sensors on the leeward side of the complex have been deactivated due to the higher measurement threshold.

The same behaviour can be seen in Figures 7 and 8, which show the optimal placement of sensors for wind coming from the south and west. For a combination of all three wind directions, the optimal sensors required to detect all release scenarios are shown in Figure 9. Note that, as expected, as the number of wind directions and release locations increases, the number of sensors required increases as well. However, only ten sensors with a measurement threshold of $c_t = 0.020$ are able to cover all cases.

5.2 Shopping mall

The second example considers the intentional release in an area representative of a typical shopping mall, with adjacent offices and hotels. The geometry definition is shown in Figure 10. The mesh consisted of approximately 0.82 Mtet. The release time was assumed to be 20 s, and the simulations were carried out for at least 600 s of real

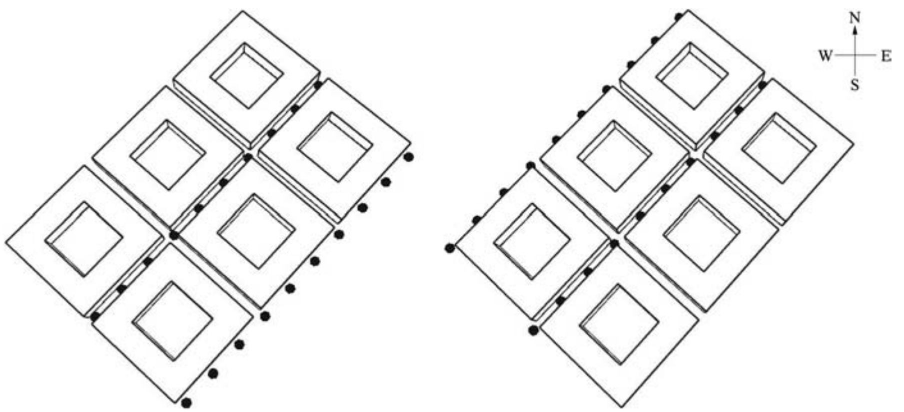

Figure 5.
Release locations for south/easterly and westerly winds

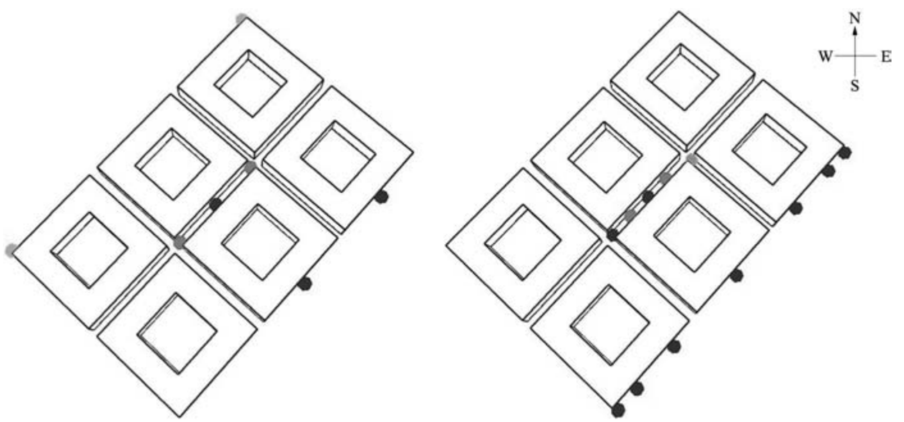

Figure 6.
Optimal sensor placement (wind: southeast) ($c_t = 0.002$, $c_t = 0.020$)

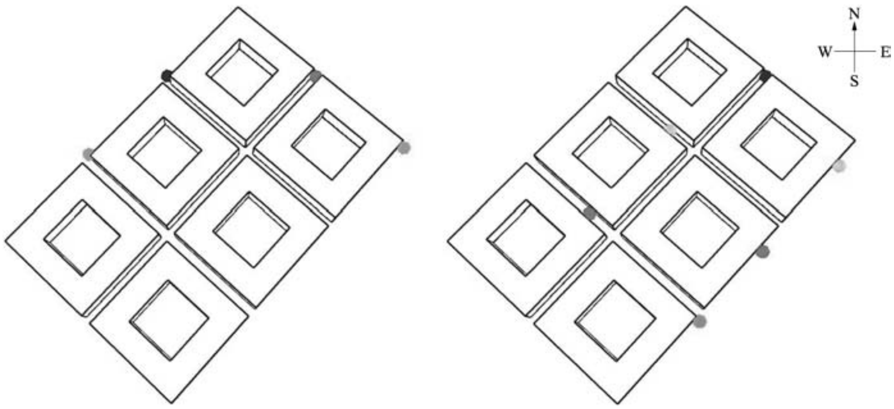

Figure 7.
Optimal sensor placement (wind: south) ($c_t = 0.002$, $c_t = 0.020$)

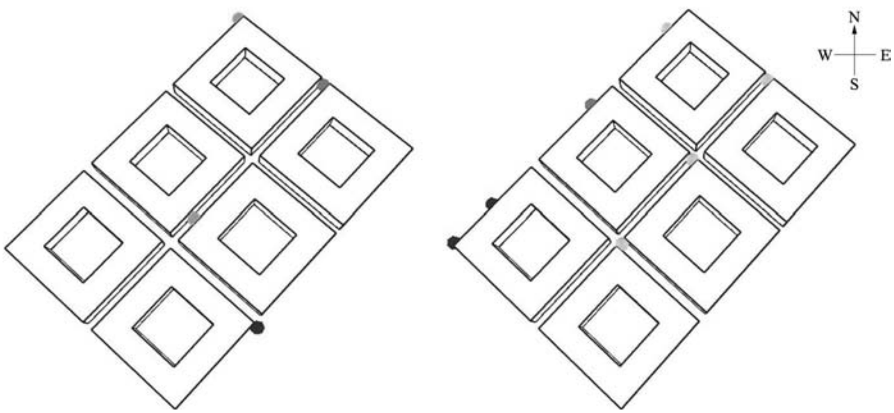

Figure 8.
Optimal sensor placement (wind: west) ($c_t = 0.002$, $c_t = 0.020$)

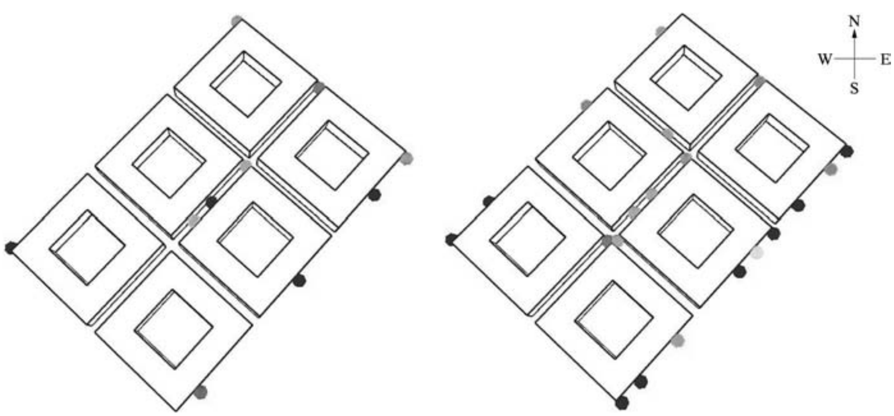

Figure 9.
Optimal sensor placement (all three wind directions) ($c_t = 0.002$, $c_t = 0.020$)

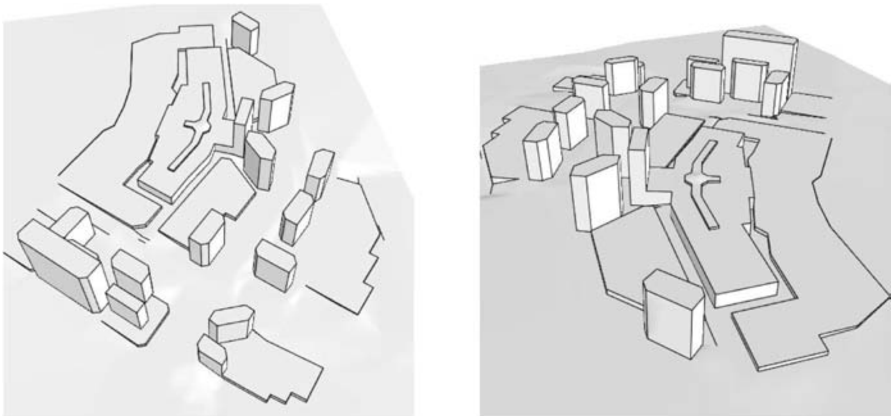

Figure 10.
Shopping mall: computational domain

time. Each of the dispersion simulation runs took between 4 and 20 min. A typical result is shown in Figures 11-13.

A total of 96 sensors, shown in Figures 14 and 15, were placed at possible locations. Figure 16 shows the 11 dispersion locations that were considered. Depending on the number of wind direction/release scenarios considered and the sensor threshold, different optimal sensor numbers and placements are obtained. Figures 17 and 18 show the optimal sensor placement (blue circles), as well as the release scenarios missed (red squares), for purely northern, western and south-western winds and a measurement threshold of $c_t = 0.10$. Note that only 1-3 optimal sensors are required to sense all measurable scenarios, but that for some cases a considerable number of release scenarios are missed. This should come as no surprise, as the sensors are in close spatial proximity while some of the release locations are further away, as well as downwind from the sensors.

For a combination of all three wind directions, the optimal sensors required to detect the maximum possible number of release scenarios are shown in Figure 19. As expected, as the number of wind directions and release locations increases, the number

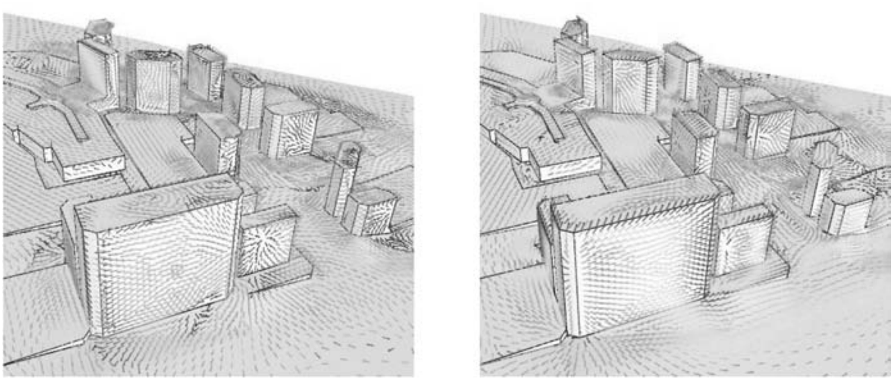

Figure 11.
Surface velocities for wind coming from north and west

of sensors required increases as well. However, only five sensors are able to cover all measurable cases.

6. Conclusions and outlook

A method for the optimal placement of sensors in order to detect the largest number of release scenarios with the minimum amount of sensors has been presented. The method is fairly general, and should be applicable to any combination of simulation

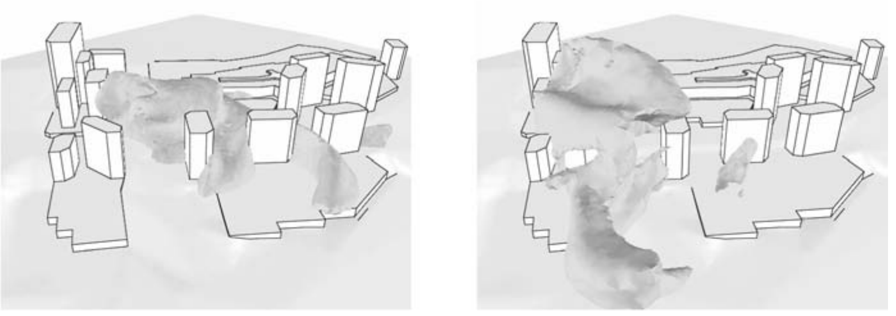

Figure 12.
Typical iso-level $C = 0.005$ cloud at $T = 200, \; 400\,\mathrm{s}$

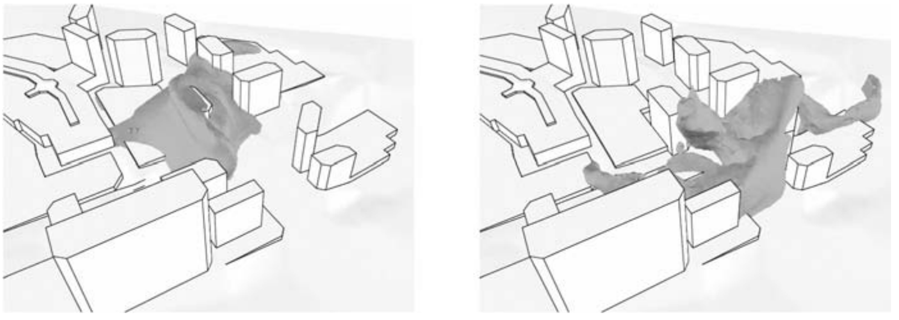

Figure 13.
Typical iso-level $C = 0.010$ cloud at $T = 200, \; 400\,\mathrm{s}$

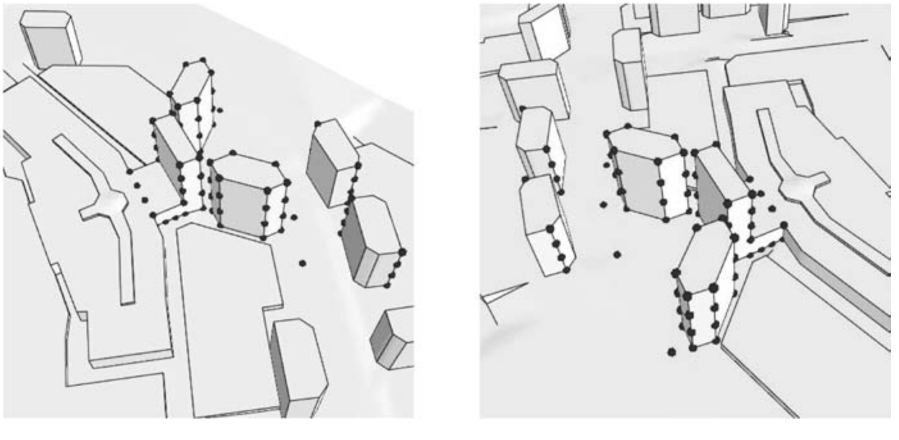

Figure 14.
Sensor placement

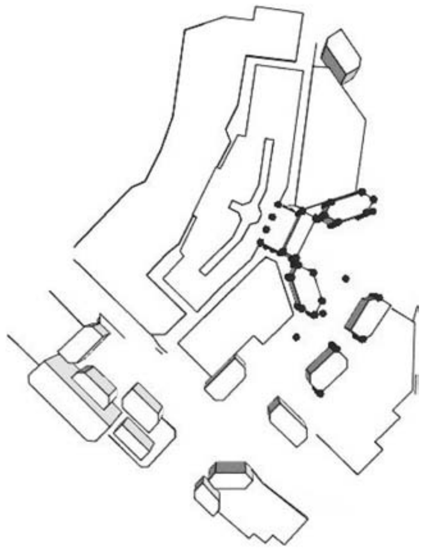

Figure 15.
Sensor placement and
release locations

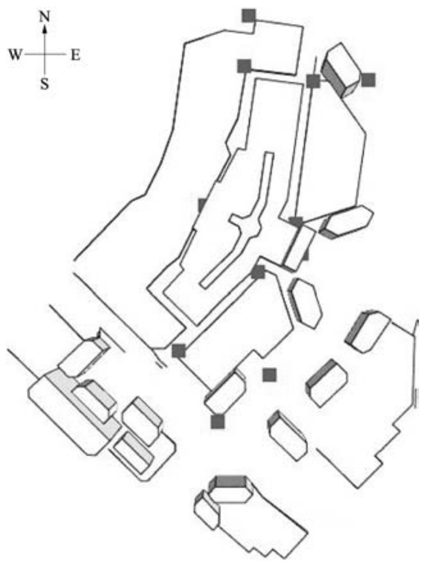

Figure 16.

codes (puff models, mass-conservation models, detailed CFD models, etc.) and release scenarios (type, amount, location, meteorological conditions, etc.). Examples showing the effect of different wind conditions and release locations demonstrate the effectiveness of the procedure. The examples also indicate that first-principles CFD models have matured sufficiently to be run in a timely manner on PCs, opening the way to optimization based on detailed physics.

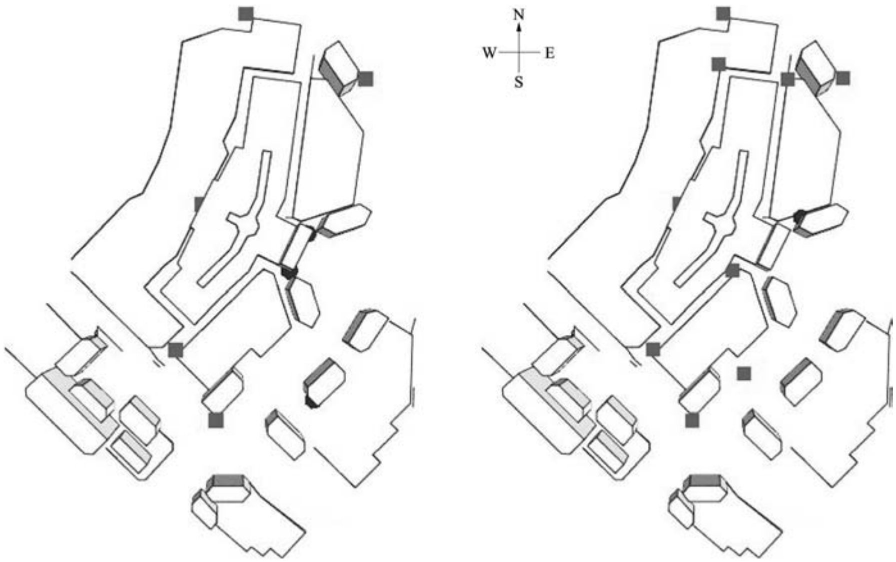

Figure 17.
Optimal sensor placement (circles) and undetected releases (squares) (winds: north, west)

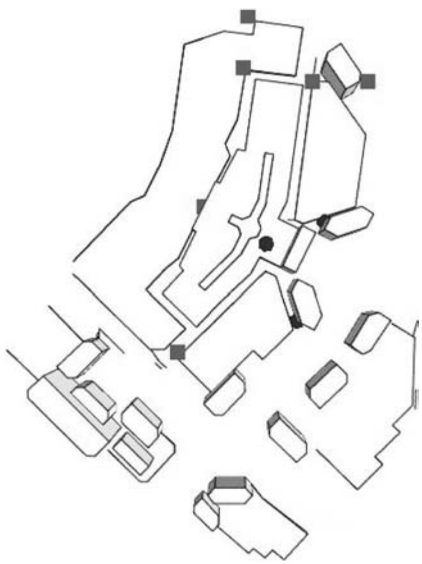

Figure 18.
Optimal sensor placement (circles) and undetected releases (squares) (wind: southwest, combination: north, west, southwest)

An obvious limitation of the present method is that the sensors are placed optimally to detect the presence of a release. In many cases, what matters is the timely (i.e. immediate) detection of a release, and it is not surprising that other techniques may be required for them (Obenshain *et al.*, 2004).

Future work will center on the extension of the method presented to these scenarios with more complex detection requirements.

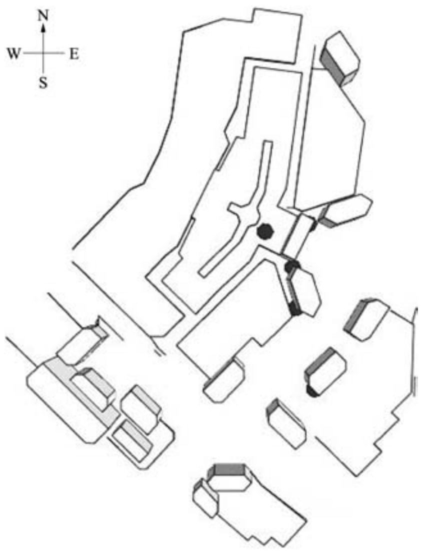

Figure 19.

References

Arya, S.P. (1998), *Introduction to Micrometeorology*, Academic Press, New York, NY.

Arya, S.P. (1999), *Air Pollution Meteorology and Dispersion*, Oxford University Press, Oxford.

Camelli, F. and Löhner, R. (2000), "Flow and dispersion around buildings: an application with FEFLO", *Proceedings of ECCOMAS 2000 Conference*, Barcelona, September.

Camelli, F., Löhner, R., Sandberg, W.C. and Ramamurti, R. (2004), "VLES study of ship stack gas dynamics", AIAA-04-0072.

Camelli, F., Soto, O., Löhner, R., Sandberg, W. and Ramamurti, R. (2003), "Topside LPD17 flow and temperature study with an implicit monolithic scheme", AIAA-03-0969.

Coirier, W.J. and Jorgenson, P.C.E. (1996), "A mixed-volume grid approach for the Euler and Navier-Stokes equations", AIAA-96-0762.

Garten, J.F., Fuechsel, P.G., Blackett, R.C., Giles, J. and Schemm, C.E. (2003), "Evaluating optimal sensor deployments for particular threat scenarios: ASSESS (APL scenario simulator for the evaluation of sensor systems)", *7th GMU Transport and Dispersion Modeling Workshop*, Fairfax, VA, June.

Hanna, S.R. (1998), "Hybrid plume dispersion model (HPDM) development and evaluation", *J. App. Meteor.*, Vol. 28, pp. 206-24.

Hanna, S.R., Briggs, G.A. and Hosker, R.P. (1982), *Handbook on Atmospheric Diffision*, NOAA DOE/TIC-11223.

Hanna, S.R., Britter, R. and Franzese, P. (2003), "A baseline urban dispersion model evaluated with Salt Lake City and Los Angeles tracer data", *Atmospheric Environment*, Vol. 37, pp. 5069-82.

Hanna, S.R., Tehranian, S., Carissimo, B., Macdonald, R.W. and Löhner, R. (2002), "Comparisons of model simulations with observations of mean flow and turbulence within simple obstacle arrays", *Atmospheric Environment*, Vol. 36, pp. 5067-79.

Kastner-Klein, P., Rotach, M.W., Brown, M.J., Fedorovich, E. and Lawson, R.E. (2000), "Spatial variability of mean flow and turbulence fields in street canyons", *Proceedings of 3rd AMS Urban Environmental Symposium*, Davis, CA, LA-UR-00-3025.

Lim, D.W., Henn, D.S. and Hookham, P.A. (2000), "Preliminary assesment of linked CFD operational dispersion models on the urban scale", *Proceedings of 3rd International Symposium on Environmental Hydraulics*, Tempe, AZ.

Löhner, R. (2001), *Applied CFD Techniques*, Wiley, New York, NY.

Löhner, R. (2004), "Multistage explicit advective prediction for projection-type incompressible flow solvers", *J. Comp. Phys*.

Löhner, R. and Camelli, F. (2004), "Mic deactivation for advection-dominated contaminant transport", *Comm. Num. Meth. Eng.*, to appear in.

Löhner, R., Yang, Chi, Cebral, J.R., Soto, O., Camelli, F. and Waltz, J. (2003), "Improving the speed and accuracy of projection-type incompressible flow solvers", AIAA-03-3991-CP.

Obenshain, K., Patnaik, G. and Boris, J.P. (2004), "Using CT-analyst to optimize sensor placement", *Proceedings of SPIE Defense and Simulation Symposium*, SPIE Paper 5416-02, Orlando, FL, April.

Prabha, T.V. and Mursch-Radlgruber, E. (1999), "Investigation of air pollution distribution in Linz: case studies to evaluate a K-type diffusion model coupled with a mass-consistent wind model", *Atmospheric Environment*, Vol. 33, pp. 4067-80.

Sherman, C.A. (1978), "A mass consistent model for wind fields over complex terrain", *J. App. Meteor.*, Vol. 17, pp. 312-9.

Sykes, R.I. and Chowdhury, B. (2004), "Source estimation from sensor alarms using SCIPUFF", paper presented at the 8th GMU Transport and Dispersion Modeling Workshop, Fairfax, VA, July.

Stern, A.C., Boudel, R.W., Turner, D.B. and Fox, D.L. (1984), *Fundamentals of Air Pollution*, Academic Press, New York, NY.

Witlox, H.W.M. (1999), "The HEGADAS model for ground-level heavy-gas dispersion – II. Time dependent model", *Atmospheric Environment*, Vol. 28, pp. 2933-45.

Appendix. Optimal sensor placement

Denoting by:

nsens: the number of sensors,

ncase: the number of release scenarios,

usens(1:nsens,1:ncase): the maximum concentration measured by the sensors for each of the cases,

uthre: the measurement threshold for concentration

the following pseudo-code summarizes the procedure:

- Initialization:
- Mark all sensors as unused: lsens(1:nsens)=0
- Mark all release cases as undetected/active: lcase(1:ncase) = 1
- Start counter for optimal sensors: nsopt = 0

- while: Release scenarios have not been recorded:
- Identify the sensor that detected the highest number of as yet undetected release cases:
- Initialize case counter: lscas(1:nsens) = 0

```
–   do icase = 1,ncase                                  !Loop over the cases
    if(lcase(icase).eq.1) then                          !Case is undetected
      do isens = 1,nsens                                !Loop over the sensors
      if(lsens(isens).eq.0) then                        !Sensor is unused
        if(usens(isens,icase).ge.uthre) then
          lscas(isens) = lscas(isens) + 1
        endif
      endif
      enddo
    endif
    enddo
–   Initialize ismax,icmax = 0
    do isens = 1,nsens                                  !Loop over the sensors
    if(lscas(isens).gt.icmax) then
      ismax = isens
      icmax = lscas(isens)
    endif
    enddo

–See if a new sensor exists, and if so: update
    if(ismax.gt.0) then                                 !A new useful sensor exists
      nsopt = nsopt + 1                                 !Update counter
      lsopt(nsopt) = ismax                              !Store optimal sensor
      lsens(ismax) = icmax                              !Mark sensor as used
      do icase = 1,ncase                                !Mark the cases covered
      if(usens(isens,icase).ge.uthre) lcase(icase) = 0
      enddo
    –endif

–endwhile
```