

Development of CPT_M3D for Multiple Chemical Plume Tracing and Source Identification

J. Sutton

Department of Computer Science
California State University, Bakersfield
Bakersfield, CA 93311, USA
jsutton@cs.csubak.edu

W. Li*

Department of Computer Science
California State University, Bakersfield
Bakersfield, CA 93311, USA
wli@cs.csubak.edu

Abstract

This paper explores the developing of a graphical simulation environment for multiple chemical plume tracing and source identification. A number of classes, including a graphical display class, make up the simulation program. Because the intent is to capture and simulate real life scenarios, the program simulates significant variation of fluid flow velocity and orientation, the significant intermittency between advection fluid flow, chemical filaments, and significant meander of chemical plumes to challenge a chemical plume tracing strategy. Thus, the advection fluid flow model class, plume model classes, sensor classes, plume tracing strategy classes, and vehicle model classes were created to bring this challenge alive in a measurable, three dimensional model. The outcome is highly promising, in that it can efficiently evaluate strategies for tracing multiple plumes, identifying their sources, and can do so even in three-dimensions.

1. Introduction

In order to develop an Autonomous Underwater Vehicle (AUV) based Chemical Plume Tracing (CPT) for natural fluid environment application, several biologically inspired CPT strategies [1]-[6] were developed. The **passive** and **active** moth-inspired CPT strategies [6] allow a single AUV to trace a chemical plume with significant meander from a potentially long distance to its initial odor source. Both strategies were implemented on a REMUS vehicle for three in-water test runs conducted in November 2002 and April 2003 at San Clemente Island, California, and in June 2003 in Duck, North Carolina [7], [8]. The accomplishment of tracing chemical plumes in

environment represents a significant step toward real applications. In real applications, however, CPT strategies for tracing multiple chemical plumes need to be investigated.

The tracking of chemical plumes in the natural environment must be much more complicated, as fluid flow, featured by turbulence, tide, and even wave, significantly challenges chemical plume tracing in near-shore ocean environments. Factors that complicate CPT include the unknown chemical source concentration, the unknown advection distance of any detected chemical, and significant spatial and temporal flow variations. Although the in-water tests on tracing multiple chemical plumes via a single REMUS vehicle in June 2003 in Duck, NC, were successful, we still need to systemically investigate strategies for tracing multiple plumes, especially, via a fleet of underwater vehicles. However, the development and evaluation of a CPT strategy are very time-consuming. Especially, the in-water tests of CPT missions via the AUVs are very expensive.

In order to evaluate the moth-inspired CPT strategies in 2D via a single robot in [6], a simulation program named CPT_S2D was developed, as shown in Fig. 1. One

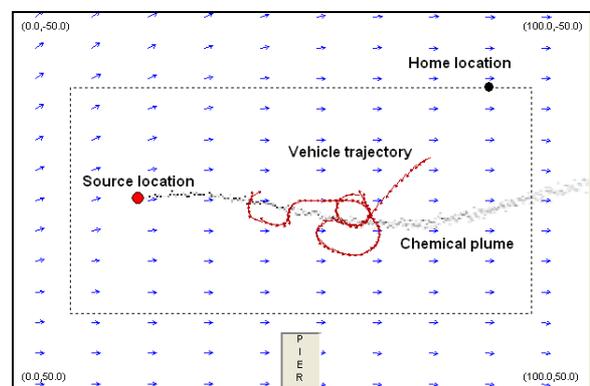


Fig. 1: Evaluation of the moth-inspired CPT strategies using CPT_S2D

* Corresponding author

disadvantage of CPT_S2D is that plume tracing simulations are limited in 2D, as the graphical display procedure in this program is based on the Win32 library. Furthermore, the plume and fluid flow models are implemented in a procedure-based design, so a significant modification is needed to simulate multiple plumes. The paper develops a new simulation program named CPT_M3D implemented in an object-oriented programming design to trace multiple chemical plumes via a fleet of robots. The new program is more powerful, as it uses a graphical accelerator library to speed up rendering time and can view the tracking of plumes in 3D.

2. Program Diagram

CPT_M3D is designed by considering the program scalability and flexibility. The graphical display uses the OpenGL library easily displaying a chemical plume in 3D. Particularly, OpenGL uses graphical acceleration to significantly speed up refresh rate. Further the graphics rendering can be offloaded to graphic hardware, reducing the CPU's overall load. The OpenGL Library automatically handles the acceleration, thus streamlining the process further. The program includes the respective classes: Plume Tracing, Environment, View, Sensor, Plume, Vehicle, and so on, as described in Fig. 2**. Relationships between the classes are established by inheritance, indicated by an up-pointing arrow, or composition, indicated by a down-pointing arrow. Plume Tracing has a composition relationship (*has-a*) to Environment and View. The View class receives all information from Environment, including vehicles' statuses, sensed information, plumes and fluid flow simulations statistics, and displays the corresponding

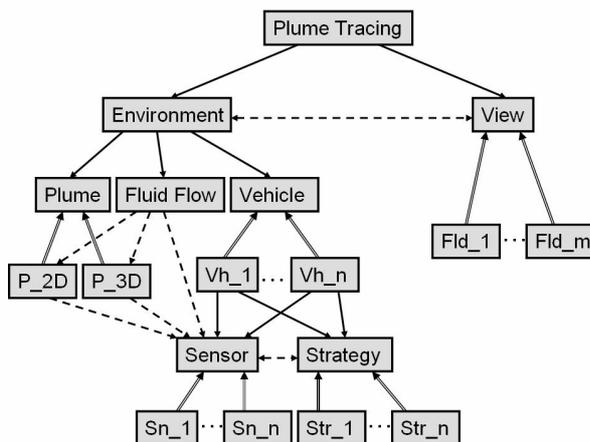


Fig. 2: CPT_M3D UML diagram

** The current implementation is slightly different from the UML diagram shown in Fig. 2 because CPT_M3D is still under revising.

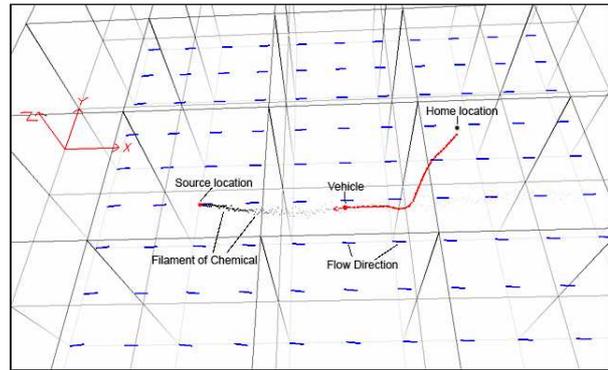


Fig. 3: A simulation test of CPT missions via CPT_M3D. The grayscale indicates plume concentrations. Short blue lines represent the fluid flow with directions from left to right. The red dot line is the vehicle trajectory. The coordinate system with the x, y, and z axes indicate the view direction of the simulation area.

fields. The View class works only for rendering, it has no knowledge of how the environment nor how a vehicle works. It gives the abstract data interfaces which is used to querying the data it uses to display. The view in all respect can be changed at anytime without affecting the other components at all. The Environment class has a composition relationship (*has-a*) to Fluid Flow, Plume, and Vehicle. The Plume class establishes an inheritance relationship (*is-a*) to P_2D or P_3D classes, which simulate 2D or 3D plumes. The Vehicle class is the base class from which a variety of vehicle models can be implemented. Currently, only the dynamics of a REMUS is immigrated from CPT_S2D into CPT_S3D for program testing. Each derived class vehicle has a composition relationship (*has-a*) to the Sensor and Strategy classes. The Sensor class establishes an inheritance relationship (*is-a*) to the classes: Sn_1, ..., Sn_n, which implement multiple types of sensors. The Sensor class, which provides the measured information, such as flow direction and chemical concentration, serves as an interface to the Strategy class. A variety of CPT strategies can be inherited from the Strategy class. Currently, one of the moth-inspired CPT strategies is implemented to navigate vehicles' maneuvers based on the information provided by the Sensor class. Fig. 3 illustrates a test run of CPT missions simulated using CPT_M3D.

In comparison with CPT_S2D implemented in procedure-based design, CPT_M3D uses the Bridge design pattern so that the Environment, Vehicle, and Sensor classes are not accessed directly by any of the other components. They are instead accessed using an interface class. The implemented interface then queries the data from the underlying data structure. This allows each component to be swapped out without the

dependence component knowing it. Modularizing the components also allow them to be contained in DLL libraries for distribution^{***}. CPT_M3D was compiled and executed on various platforms and systems, including: an AMD Athlon 64 X2, running Windows XP, compiled using MS VS v6. A Macbook Pro, Intel Core 2 Duo, compiled using Xcode 2.5. A Parallels Virtual machine, running Windows XP, compiled using MS VS v6. CPT_M3D library dependencies include C++ STL, OpenGL and GLUT.

3. Models

CPT_M3D upgrades the procedure-based design version of the filament-based plume model and an advective fluid flow model [10] from CPT_S2D into object-oriented design version, and integrates the moth-inspired CPT strategies [6] and [9].

A chemical released at a source location will be manipulated by turbulent and molecular diffusion while transported advectively by the wind. The odor dispersion process is dominated by turbulent dispersion. Turbulent dispersion involves a wide range of length (or eddy) scales. Eddies larger than the scale of a section of the plume (i.e., a puff) transport the puff as a whole, causing the ensemble of puffs to appear as a sinuous plume. Eddies smaller than the puff mix the components of the puff causing little puff motion or growth. Eddies on the order of the puff size cause significant growth/distortion of the puff and motion of individual filaments relative to the instantaneous plume centerline. The advective fluid flow model is described by a group of partial differential equations and is numerically implemented. The very important issue addressed in our simulation program is to simulate the significant filament intermittency, which greatly challenge a CPT strategy, as shown in Fig. 4.

The Vehicle class encapsulates an important virtual member function $\tau = \Phi(\ddot{x}, \dot{x}, x)$. Each derived will overwrite this function to implement an actual vehicle dynamics, for example, the REMUS class overwrites this function by implementing the REMUS vehicle dynamics to evaluate the moth-inspired CPT strategies for the in-water tests. Similarity, the Sensor class encapsulates a virtual function to describe dynamic response of a sensor. In our case, a second-order low-pass filter implemented the chemical sensor transit time response. In fact, such a time response from chemical signals to electrical signals can be ignored in comparison with the mechanical inertia of an AUV.

CPT_M3D integrates the moth-inspired CPT strategies, consisting of four fundamental behaviors: pluming finding, plume tracing, plume reacquiring, and

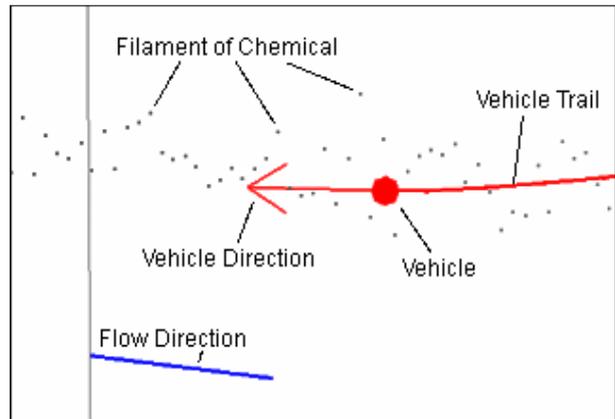


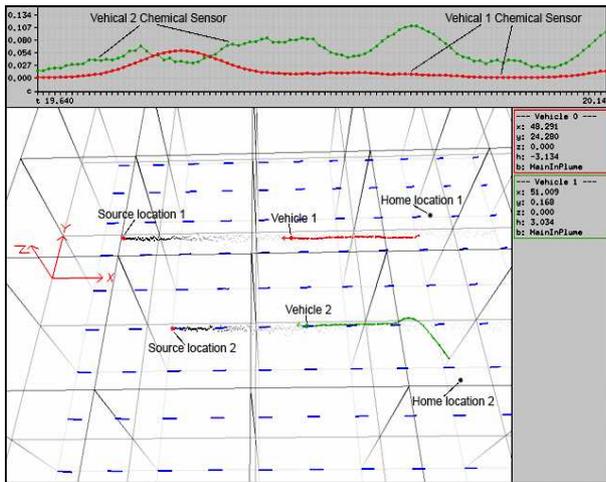
Fig. 4: An expanding scale illustrates significant filament intermittency, which causes the vehicle to loss contact with chemical. The flow direction represented by a blue line points from left to right.

source declaration (identification). The plume tracing and reacquiring are abstracted from male moths searching and tracing pheromone to a female moth location, which is summarized as follows. When a moth detects pheromone, it tries to maintain contact with the plume and to move upwind towards the source location. The maneuver is exhibited as a short sprint predominantly in the upwind direction. Repeated pheromone encounters result in the moth progressively approaching the chemical source. When a moth has not detected pheromone for a sufficiently long period of time, it ceases upwind movement and performs progressively widening crosswind excursions, termed 'casting'. In this case, the moth appears to be searching for pheromone near the position where it was last detected. This reacquiring behavior can continue for several seconds until either chemical is again detected or the moth behavior changes. The work in [7] used a subsumption architecture (its extended version presented in [11] integrated the obstacle avoiding and map construction behaviors) to coordinate these behaviors according to an inhibition or a suppress mechanisms.

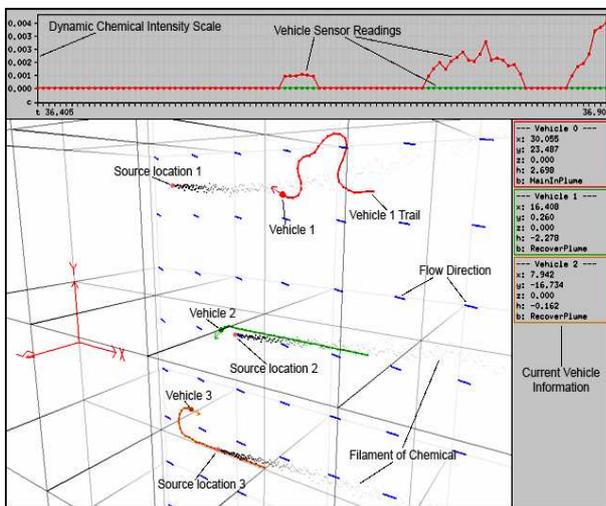
4. Simulation Examples

This section presents several examples of tracing multiple plumes via multiple robots. The simulated plume models achieve significant computational simplification relative to turbulence models, but it was designed to maintain the plume characteristics that significantly complicate the plume tracing problems (intermittency, meander, and varying flow) caused by natural fluid flow. Instead of adjusting the Reynolds numbers, it controls a filament release rate (5–10 filaments/s) to simulate filament intermittency, as shown in Fig. 4. Especially, this

^{***} The CPT_M3D program (for Windows or any Unix platform that supports OpenGL & GLUT) will be available for download at <http://www.ee.ucr.edu/~wli> through the corresponding author.



(a): The tracking of two plumes via two robots.



(b): The tracking of three plumes via three robots.

Fig. 5: Simulation studies of the tracking of multiple plumes via multiple robots. Vehicles (not to scale) and plumes within the simulated search area. The X coordinate is between 0 and 100 m and is positive in the mean downwind (i.e., horizontal) direction. The Y coordinate is between -50 and 50 m and is positive up. Heading is measured positively with respect to the Z-axis in a right-handed coordinate system. The grayscale indicates above threshold concentration. The blue solid lines indicate local wind vectors, whose directions in average point from left to right.

model addresses the meandering nature of the plume and manipulates flow variation as the key factors to challenge the CPT strategies. In order to generate a significant plume meander, an operation area must be big enough. Therefore, the simulation environment is defined in the following way: first, An operation area covers an area of

[0,100]×[-50,50] m. Second, the filament release rate is 5 filaments/s because this low release rate results in significant plume intermittency. This often causes the vehicle to lose contact with the plume and consequently to make spurious identification. Third, the mean fluid velocity is 1 m/s. Those flow speed and flow variation, which are much larger than those detected during the in-water test runs, disperses the chemicals rapidly to challenge the efficacy and robustness of the plume-tracing strategies. Next, measured flow direction is corrupted by additive noise from a white normal random process with a significant variation.

CPT_M3D is able to simulate as many plumes and vehicles as we need. The number and locations of plume sources can arbitrarily selected. The source locations are unknown to any vehicle during CPT test runs, but allow us to check the accuracy of an identified source location. Fig. 5 illustrates the simulation screens, consisting of three windows: top window (chemical reading window), right window (vehicle status window), and main window (operation area).

The chemical reading window displays the chemical concentrations detected by each vehicle. The axis on the left side of the view dynamical adjusts to the maximum chemical sensor reading within a sliding two second time window (this is simulation time, not real time). The bottom axis, naturally, slides along the time line, with current time to the far right and two seconds in the past to the left. The line and points displaying the chemical intensity readings for a particular vehicle's sensor takes the given color of that vehicle.

The vehicle status window shows the current information on each vehicle, including its active behavior, the current location and orientation of each vehicle.

The main window in the middle of the simulation screens displays the operation area in which there are vehicles, and plumes, etc****. The main window can display the CPT missions in 3D. The view in the main window can be zoomed, moved, panned and tilted, using either the mouse and/or a keyboard. The simulation timer runs at a constant rate, while the display runs on it's own timer allowing for machines without hardware graphics acceleration to simulate at a compliable speed. The simulation timer rate can also be increased, decreased at anytime, using the keyboard. The simulation timer rate can be preset in the settings file (see below). The simulation time can also be stopped and started anytime using the keyboard. The simulation timer runs at a constant rate, while the display runs on it's own timer allowing for machines without hardware graphics acceleration to simulate at a compliable speed. The simulation timer rate can also be increased, decreased at

**** A color printing can display the maneuver of each vehicle clearly.

anytime, using the keyboard. The simulation timer rate can be preset in the settings file (see below). The simulation time can also be stopped and started anytime using the keyboard. Being that the simulation timer and graphics render timer are separate, if the simulation environment timer is stop the 3D can still be zoomed, moved, panned and tilted.

Fig. 5(a) illustrates a scenario of CPT mission with two plumes and two vehicles. The locations of the plume odors are chosen at $(20, 0, 0)$ and $(5, 25, 0)$, respectively. One vehicle starts at $(80, 30, 0)$, and the other at $(80, -15, 0)$. The plume finding behavior drives both of the vehicles to cross the operation area for searching the plumes. The red dot line indicates the trajectory of vehicle 1, while the green dot line indicates the trajectory of vehicle 2. Once they detect the chemical concentrations above thresholds, they activate the Maintain Plume behaviors to trace the chemical plumes in the up-flow direction, as indicated by "MainInPlume" state in the right window. The chemical concentrations detected by the both vehicles are drawn by the corresponding colors in the chemical reading windows.

Fig. 5(b) illustrates a scenario of CPT mission with three plumes and three vehicles. The locations of the plume odors are chosen at $(20, 0, 0)$, $(5, 25, 0)$, and $(15, -23, 0)$, respectively. The home locations for the vehicles are specified at $(80, 30, 0)$, $(80, -15, 0)$, and $(60, -25, 0)$. The three vehicles start their CPT maneuvers with the Find Plume behaviors at their home locations. They trace the chemical plumes in up-flow directions once they detect the chemical plume concentrations above thresholds. The trajectory of vehicle 1 (the red dot line) indicates that its maneuver includes the Maintain Plume and Reacquire Plume behaviors. The chemical plume tracing maneuvers of vehicles 2 and 3 are represented by the green and orange trajectories. Vehicles 2 and 3 already reach the sources' locations of the plumes which they are tracing and activate the Declare-Source behavior to identify the sources locations.

CPT missions simulated in Fig. 5(a) and Fig. 5(b) runs the same excludable program, but it uses two different settings in a setting file. The setting file eliminates the need to recompile for various types of simulations. The setting file contains various parameters that be read when the simulation program starts. CPT_M3D supports up to N number of vehicles, whose home locations and colors can be predefined through the setting file. CPT_M3D additionally supports up to M number of chemicals plumes, whose source locations can be predefined through the setting file. The remaining parameters modify fluid flow, simulation timing, strategy, object rendering color, etc.

5. Conclusions and Further Research



Fig. 6: A fleet of Surveyor's SRV-1 internet-controlled robots with wireless communication is chosen for the tracing of multiple plumes.

This paper presents CPT_M3D simulation program and adopt the moth-inspired CPT strategy to trace multiple chemical plumes via multiple robots. In order to trace multiple chemical plumes efficiently, additional coordination mechanisms to the moth-inspired CPT strategies will be developed to automatically determine the size of a robot fleet and to assign their maneuvers' activities through interactively communication between each individual robot.

Next, the obstacle class presented in [12] will be integrated from CPT_S2D into CPT_M3D to create obstacles in the operation area. Also, the fuzzy following-obstacle behavior [13] will be integrated in CPT_M3D.

For experiments on CPT via a fleet of real robots, a communication class setting up a communication protocol will be developed to get communications between multiple robots and the host computer. At the next step, we will use a fleet of Surveyor's SRV-1 internet-controlled robots, as shown in Fig. 7, to test CPT strategies for tracing multiple plumes. These robots are equipped with a digital video camera with resolution from 160×28 to 1280×1024 pixels, laser pointer ranging, and WLAN 802.11b/g networking. They can operate as a remotely-controlled webcam or a self-navigating autonomous robot. These robots can be used for our indoor or outdoor experiments. Another advantage of a SRV-1 robot is that the manufacturer offers open source code and expandable architecture, to clients, so that it is easy to integrate a flow detector and a chemical sensor on the robot through the I2C or GPIO expansion bus signals.

The moth-inspired CPT strategies can be extended to trace a chemical plume and identify the odor source with static location in a 3D space. In order to evaluate the performance of the moth-inspired CPT strategies in 3D dimension, for example, the tracking of chemical plumes with anisotropic diffusion, a 3D plume model is under development and being integrated into CPT_M3D.

6. Acknowledgements

This work was supported in part by the RCU grant of CSUB.

7. References

- [1] E. A. Cowen and K. B. Ward, "Chemical Plume Tracing," *Environmental Fluid Mechanics*, vol. 2, no. 1-2, pp. 1-7, 2002
- [2] F. W. Grasso and J. Atema, "Integration of flow and chemical sensing for guidance of autonomous marine robots in turbulent flows," *Journal of Environmental Fluid Mechanics*, vol.1, pp.1-20, 2002.
- [3] Q. Liao and E. A. Cowen, "The information content of a scalar plume – A plume tracing perspective," *Environmental Fluid Mechanics*, vol. 2, no. 1-2, pp. 9-34, 2002.
- [4] M. J. Weissburg *et al.*, "A multidisciplinary study of spatial and temporal scales containing information in turbulent chemical plume tracking," *Environmental Fluid Mechanics*, vol. 2, no. 1-2, pp. 65-94, 2002.
- [5] A. T. Hayes, A. Martinoli, and R. M. Goodman, "Distributed chemical source localization," *IEEE Sensors Journal*, vol. 2, pp. 260-271, 2002.
- [6] W. Li, J. A. Farrell, and R. T. Cardé, "Tracking of fluid-advected chemical plumes: Strategies inspired by insect orientation to pheromone," *Adaptive Behavior*, vol.9, pp.143-170, 2001.
- [7] W. Li, J. A. Farrell, S. Pang, and R. M. Arrieta, "Moth-inspired chemical plume tracing on an autonomous underwater vehicle", *IEEE Transactions on Robotics*, vol.22, no.2, pp.292-307, 2006.
- [8] J. A. Farrell, S. Pang, and W. Li, "Chemical plume tracing via an autonomous underwater vehicle," *IEEE Journal of Ocean Engineering*, vol.30, pp.428-442, 2005.
- [9] W. Li, "Abstraction of odor source declaration algorithm from moth-inspired plume tracing strategies," in *Proc of the 2006 IEEE International Conference on Robotics and Biomimetics*, pp. 1024-1029, 2006.
- [10] J. A. Farrell, J. Murlis, X. Long, W. Li, and R. Carde: "Filament-Based Atmospheric Dispersion Model to Achieve Short Time-Scale Structure of Chemical Plumes," *Environmental Fluid Mechanics*, vol.2, pp.143-169, 2002.
- [11] W. Li, M. M. Elgassier, C. Bloomquist, and K. Srivastava: "Multisensor integration for declaring the odor source of a plume in turbulent fluid-advected environments," *Proc. of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'06)*, pp.5534-5539, 2006.
- [12] W. Li, and D. Carter: "Subsumption architecture for fluid-advected chemical plume tracing with soft obstacle avoidance," *Proc. of the Ocean Marine Technology and Ocean Science*, 2006.
- [13] W. Li, and J. Zhang: "Moth-inspired chemical plume tracing by integration of fuzzy following-obstacle Behavior," *Proc. of the 2008 IEEE World Congress on Computational Intelligence*, 2008 (in press).