

Modélisation du comportement humain pour la simulation d'évacuation de bâtiment en feu

Julien Valentin¹, Florent Coudret², Eric Gouardères¹ et Wilfrid Lefer¹

¹LIUPPA - Université de Pau et des Pays de l'Adour

²Centre Scientifique et Technique du Bâtiment, Sophia-Antipolis

Résumé

La simulation d'évacuation de bâtiment en proie au feu a pour objectif de diagnostiquer, dès les premières phases de la conception d'un bâtiment, sa capacité à être évacué par des individus dans différents scénarios d'incendie. De telles simulations peuvent servir de base de décision pour l'implantation de dispositifs de sécurité, telles que des systèmes d'extraction de fumée, la signalisation des issues ou encore des extincteurs, la révision de la conception des bâtiments ou le développement de procédures d'évacuation propres à ces derniers. La prise en compte du facteur humain est essentielle afin d'espérer des résultats réalistes. Dans cet article nous décrivons une architecture logicielle et un ensemble de concepts permettant de contrôler l'adaptation des comportements humains dans le cadre d'un scénario d'évacuation de bâtiment en feu.

Simulating the evacuation of building in fire aims at evaluating, at the early stages of building design, the ability of the building to allow safety evacuation by its occupants, in various evacuation scenarios. Such simulations may serve as guidelines for populating the building with safety devices, such as smoke extractors, signage or extinguishers, building design, or the definition of evacuation rules. Taking into account human behaviour is essential to obtain realistic results. This paper describes a software architecture together with a set of concepts for the design of adaptive human behaviours for the simulation of building in fire evacuation scenarios.

Mots clé : Animation comportementale, simulation de foule, modélisation microscopique, situations d'urgence, sécurité, bâtiment en feu.

1. Introduction

Les outils de simulation des divers phénomènes naturels, tels que incendies, ouragans et autres tremblements de terre, appliqués à des modèles numériques des constructions humaines sont aujourd'hui de plus en plus utilisés afin de détecter les défauts de conception avant même la construction effective des ouvrages. La visualisation des résultats de ces simulations sert de base de communication visuelle entre la Maîtrise d'Ouvrage et les métiers de la conception. Elle permet d'établir un diagnostic visant à améliorer la sécurité des lieux. La simulation d'évacuation de bâtiments en cas d'incendie fait partie de ces outils mais présente la particularité de se fonder sur un mécanisme très difficile à cerner : le comportement de ses occupants en situation d'urgence. En effet, comme [PBPD02] le fait remarquer, aucune théorie ne fait l'unanimité dans le domaine des sciences cognitives car :

- on manque de données fiables sur les interactions sociales entre individus lors de situations d'urgence,

- les chercheurs en modélisation du comportement humain sont peu enclins à fournir un modèle général qui les enfermerait dans des hypothèses réductrices.

La plupart des modèles de simulation postulent la totale rationalité de l'humain et simulent la diversité des individus en les dotant de propriétés physiques, physiologiques et de tâches à effectuer, différentes et prédéterminées, l'ensemble de ces caractéristiques étant appelé archétype de comportement. Dans sa théorie de la rationalité limitée, Herbert Simon propose une autre vision : le comportement humain ne s'expliquerait pas uniquement par un quelconque ensemble de propriétés mais serait aussi lié aux limitations de la rationalité humaine en termes de moyens cognitifs et d'informations à sa disposition [Par05].

Cet article présente le développement d'une architecture microscopique pour la simulation de l'évacuation de bâtiments en proie au feu. L'originalité de ce travail réside dans :

- un modèle microscopique : le mécanisme de raisonnement de chaque personne est modélisé individuellement, via un archétype de comportement, le comportement de foule n'étant que l'émergence de l'ensemble des comportements individuels,

- une gestion de la connaissance et de la croyance au cours du temps, aboutissant à l'absence d'omniscience, contrairement à ce qui est observé dans les modèles existants,
- une implémentation du principe de rationalité limitée qui aboutit à des comportements différents d'individus pourtant dotés du même archétype comportemental et disposant de la même connaissance de leur environnement : ce principe permet notamment de modéliser le phénomène de panique, essentiel pour la compréhension des scénarios en situations d'urgence.

La suite de cet article commence par un état de l'art de la simulation comportementale et une évaluation des principaux simulateurs d'évacuation existant. Puis notre architecture d'adaptation comportementale est présentée et notre modèle de décision décrit en détails. La dernière partie de l'article présente une implémentation GPU du système de planification individuel. Enfin nous concluons en résumant les propriétés de notre architecture.

L'objectif de ce travail étant d'évaluer la sécurité des bâtiments, le résultat des comportements observés importe davantage que la qualité du rendu. Nous concentrerons donc notre étude sur l'aspect comportemental. Une optimisation de l'algorithme de planification, tirant profit des architectures GPU, est proposée à la section 5 afin de réduire le temps de calcul de la solution mais n'a pas pour objectif de simuler en temps réel. Le lecteur pourra trouver dans [Tha08] pour un état de l'art complet sur le rendu temps réel de foule.

2. Simulation comportementale

La simulation comportementale consiste à tenter de faire reproduire par des entités virtuelles les mécanismes décisionnels qui mènent à l'action, en prenant en compte un objectif et les contraintes imposées par leur environnement. Il existe deux grandes classes de modèles de simulation comportementale : les modèles multi-agents [Fer97] et les modèles physiques, dont il a été montré qu'ils sont également bien adaptés à la simulation de grandes foules [HLTC03]. Dans la simulation d'évacuation de bâtiment, le problème est défini spatialement et peut alors se résumer en la recherche des meilleurs déplacements à entreprendre pour satisfaire l'atteinte d'une sortie en fonction des contraintes de mouvement.

De nombreux systèmes de simulation de foule distinguent deux grandes classes de comportements :

- les comportements navigationnels, qui permettent de satisfaire la prise en compte de contraintes de mouvement de manière instantanée ou à court terme,
- les comportements planifiés, qui permettent de satisfaire l'atteinte d'un objectif en intégrant les contraintes de mouvement.

Remarquons une différence fondamentale entre ces 2 classes de comportements, liée à la notion de but : les comportements planifiés sont nécessairement associés à un but

à atteindre, alors que les comportements que nous appelons navigationnels peuvent avoir pour seul objectif de satisfaire des contraintes de mouvement, telles qu'éviter les obstacles par exemple. Ces comportements peuvent tous être modélisés par chacune des deux grandes classes de modèles mentionnées ci-dessus.

2.1. Comportements navigationnels

Les comportements navigationnels constituent une catégorie des comportements réactifs de la terminologie multi-agents [Fer97]. Ils permettent d'adapter le comportement d'un agent en fonction des seules contraintes locales de mouvement et n'intègrent aucune notion de but. Ils sont souvent implémentés en tant que force appliquée à l'agent, qui le contraint à se déplacer, cette force étant la résultante d'une formule mathématique comportant divers paramètres.

Lorsque les seuls paramètres sont la position et la distance, ce type de comportement est basé sur la description du comportement observé et n'exprime aucune forme d'anticipation de la trajectoire. Les équations proposées par Helbing dans [HMFB00] constituent un exemple typique de ce modèle. Le fait d'y ajouter la vitesse permet de doter les individus d'une forme d'anticipation puisqu'on va pouvoir tester l'intersection non plus simplement de la position mais de l'extrémité du vecteur vitesse, ce qui aura pour effet que les piétons vont anticiper la présence des obstacles et modifier leur trajectoire à distance respectable de ceux-ci [Rey06] [WvdBM09]. Enfin le modèle appelé *Steering inverse* [AOM03] généralise cette formule pour en faire une fonction de coût, intégrant éventuellement des paramètres non physiques, et considère toutes les évolutions possibles de l'état de l'agent - enchaînements de comportements atomiques - afin de sélectionner la meilleure au sens de la fonction de coût.

2.2. Comportements planifiés

Les comportements planifiés cherchent à reproduire le raisonnement qui mène à la décision par déduction. Ils consistent typiquement en la recherche du plus court chemin menant à un but au sens d'une certaine fonction de coût, qui peut prendre en compte de nombreux paramètres représentant les différentes contraintes de mouvement. La résolution de ce problème nécessite généralement de discrétiser l'espace afin de rechercher une solution approchée, sous la forme d'une grille régulière, d'un graphe ou d'une surface triangulée.

Dans le cas d'une grille régulière, les arêtes de celle-ci sont valuées par le coût de passage du centre d'une des cellules incidentes au centre de l'autre cellule incidente, via l'utilisation d'une structure de type MAC [FSJ01]. [TCP06] décrit une technique permettant la résolution du problème pour des groupes d'agents partageant buts et vision de l'environnement, basée sur la Fast Marching Method [SV03]. [SOB08] propose une implémentation sur GPU d'une variante de cette

	Modèle de comportement	Intégration incendie	Hypothèses fondatrices invalides
Evac/FDS	Hybride : modèle de fluide + navigationnel	Oui	Omniscience Connaissance des issues connues des autres
Massegress	Purement réactif	Non	Classification des comportements types
Hidac	Hybride : A^* + Steering	Non	Omniscience
Exodus	Cognitif	Oui	Omniscience
Pathfinder	Hybride : A^* + Steering inverse	Non	Omniscience+Intention fanatique
EveLife	Hybride : FIM modifiée	Oui	Néant

Table 1: Tableau récapitulatif des principaux simulateurs d'évacuation existant ainsi que notre simulateur EveLife.

technique - la Fast Iterative Method [JW07] - mieux adaptée aux architectures de traitement parallèle. [PJS06] propose le E^* , un algorithme intégrant la réparation locale introduite par le D^* décrit ci-dessous à la Fast Marching Method. [HB08] propose de structurer hiérarchiquement l'espace en fusionnant les cellules adjacentes de même coût afin de limiter la complexité algorithmique de la méthode.

L'autre approche consiste à plaquer un graphe sur l'environnement, les seules positions praticables étant alors les noeuds du graphe. Les arêtes du graphe sont valuées par le coût du passage d'un noeud au noeud voisin. Le problème revient alors à trouver la suite ordonnée de noeuds minimisant la somme des coûts des arêtes les reliant entre eux. L'algorithme de Dijkstra est le premier et le plus connu permettant de calculer un tel chemin. L'algorithme A^* s'en est distingué par l'utilisation d'une évaluation heuristique dirigeant la recherche. Puis le D^* et le LPA^* [LKL02] ont introduit la réparation locale du chemin.

Nous avons opté pour une structure en grille régulière car elle permet d'obtenir des mouvements plus fluides, puisque les positions occupées par les individus ne sont pas limitées aux noeuds du graphe, ce qui rend en particulier la fluidité des trajectoires indépendante de la résolution de la discrétisation, mais pas leur précision évidemment. L'utilisation d'une discrétisation sous forme de graphe requiert de plus une phase de positionnement des noeuds du graphe, qui doit répondre à un souci d'optimisation des trajectoires, mal adapté à une géométrie de l'espace praticable qui va être altérée par la présence du feu et de la fumée. Enfin, une grille régulière peut être facilement définie et adaptée à une géométrie de bâtiment, comportant notamment des sols plans.

2.3. Etude des modèles existants

Notre objectif n'est pas de créer des mouvements de foule seulement réalistes en apparence, comme par exemple dans les applications de jeux vidéo où l'objectif est de renforcer le réalisme de l'environnement ludique en le peuplant d'individus dont les mouvements doivent paraître naturels. Ce qui nous intéresse, c'est de mettre en évidence la variété des comportements humains que l'on peut observer dans une situation d'évacuation de bâtiment en feu et de produire des

comportements individuels et réalistes. C'est la fiabilité des simulations, sur le plan comportemental, qui importe, plus que l'aspect esthétique des résultats.

Dans ce paragraphe nous ne considérons que les modèles spécifiquement conçus pour l'évacuation de bâtiment en feu. Comme le souligne X. Pan [Pan06], les êtres humains ne se comportent pas tous de la même façon lors des situations de crise et ces comportements ne peuvent simplement s'expliquer par la différence des tâches qu'ils ont à accomplir. La manière de se comporter dans ces situations est fortement dépendante de l'individu et par conséquent l'adaptation individuelle des comportements est un critère de première importance dans ce type de simulation.

Pour Evac/FDS [Hel07], Massegress [Pan06], Pathfinder[†] et Hidac [PAB07], la prise de décision résulte d'un calcul mathématique faisant intervenir divers paramètres, le résultat étant un vecteur déplacement. L'espace est donc continu au sens où n'importe quel point peut être atteint. A l'inverse Exodus [VGL09] travaille sur une version discrète de l'espace, concrètement un graphe est plaqué sur l'espace et seuls les noeuds du graphe constituent des positions possibles des agents, les déplacements se faisant de noeud en noeud en suivant les arêtes du graphe. Exodus représente le travail le plus abouti et est un produit commercial.

Les défauts de ces systèmes viennent des hypothèses simplificatrices sur le comportement qu'ils sont censés imiter, en particulier l'omniscience, qui suppose que la topologie du bâtiment est connue de tous. Mais aussi l'intention fanatique, signifiant que le choix de la sortie pressentie ne sera jamais remis en question, que l'on trouve dans Pathfinder. Quant à Massegress, il limite l'éventail des comportements exprimables à un ensemble prédéfini. Enfin, le modèle de décision individuel de Evac/FDS implique que les issues pressenties par les autres agents en présence sont connues de chaque agent, ce qui a pour effet de faire émerger une répartition des agents sur ces issues, ce qui constitue une hypothèse très optimiste dans une situation où les communications se révèlent difficiles.

Un résumé des techniques, fonctionnalités et critiques de ces

[†]. <http://www.thunderheadeng.com/pathfinder>

Sens	Percept	Contrainte	Comportement
Vue	Feu	Spatiale	$-\nabla CDist(CDist_{cur})$
Vue	Obstructions	Spatiale	$-\nabla CDist(CDist_{cur})$
Vue	Espace déambulable	Spatiale	$-\nabla CDist(CDist_{cur})$
Vue	Fumée	Toxicité	$-\nabla CTox(CTox_{cur})$
Toucher	Température	Chaleur	$-\nabla CTemp(CTemp_{cur})$
Vue	Autres individus	Densité	$-\frac{k}{\gamma} \vec{n}$
Vue	Panneaux d'aide	Aide	$+\frac{k}{\gamma} \vec{n}$

Table 2: Les percepts, le sens permettant leur détection, les contraintes navigationnelles associées et le comportement résultant. Le signe donne la nature de la force exercée : + pour une force attractive, - pour une force répulsive.

systèmes est donné dans la table 1. Une description plus détaillée a été mise à disposition par le NIST [KD05].

3. Architecture de simulation

Nous travaillons sur la simulation de l'évacuation de bâtiment en feu dans le cadre d'une collaboration avec le CSTB[‡] de Sophia Antipolis. Les modèles numériques développés par le CSTB sont des modèles au format IFC - *Industry Foundation Classes* - conçus à l'aide du logiciel Archicad[§] ou encore des modèles uniquement géométriques réalisés par des modeleurs tels que Blender. Les différents bâtiments sont visualisés à l'aide de la bibliothèque OpenSceneGraph[¶]. Les scénarios d'incendie sont simulés grâce au logiciel de propagation de feu et de fumée FDS - *Fire Dynamics Simulator*^{||} - mis à disposition par le NIST - *National Institute of Standards and Technology*. Ces deux outils représentent les contraintes technologiques de l'implémentation réalisée.

La figure 1 montre l'organisation globale des traitements et les flux de données associés, en distinguant les modules fournis par le partenaire industriel, celui qui est réalisé par un logiciel tiers et ceux qui sont une contribution objet de cet article.

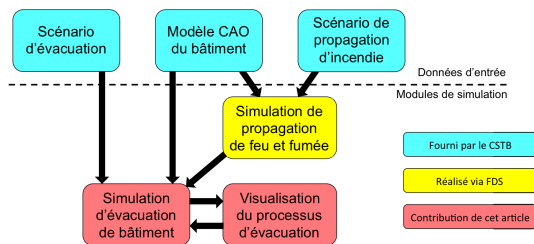


Figure 1: Architecture générale de notre simulateur d'évacuation de bâtiment en feu.

[‡]. Centre Scientifique et Technique du Bâtiment

[§]. <http://www.abvent.com/software/archicad>

[¶]. <http://www.openscenegraph.org>

^{||}. <http://www.fire.nist.gov/fds>

3.1. Simulation de propagation d'incendie

La simulation d'incendie est un processus très coûteux, qui peut parfois nécessiter plusieurs jours de calcul. Elle s'effectue donc en amont de la simulation de l'évacuation. Exécuter des simulations de scénarios d'incendie avec le logiciel de simulation FDS nécessite de convertir la géométrie des bâtiments en voxels car FDS utilise une description volumique. Les maquettes du bâtiment sont d'abord épurées afin de n'en garder que la description géométrique. L'étape suivante consiste à introduire manuellement les méta-informations correspondant aux différents matériaux constitutifs du bâtiment. Techniquement, ces informations sont encodées dans l'attribut couleur des différentes primitives géométriques de la maquette épurée, à l'aide d'un modeleur 3D. La maquette subit alors une discrétisation en voxels à la résolution requise pour la simulation d'incendie. Un traitement consistant à fusionner les cellules adjacentes de même matériau est ensuite appliqué, afin de créer des pavés homogènes et ainsi réduire le nombre de cellules et donc la complexité de la simulation de propagation du feu. Enfin le fichier d'entrée de FDS est généré en associant à chaque cellule son matériau.

Le résultat de la simulation FDS est constitué des densités en feu et fumée, de la température et de la toxicité, donnée en FED - *Fractional Effective Dose* [Spe96] - pour chaque cellule à chaque pas de temps de la simulation.

3.2. Simulation d'évacuation

Chaque étage du bâtiment est représenté par plusieurs grilles régulières 2D :

- une grille statique des obstacles,
- une grille statique contenant les panneaux d'aide à l'évacuation,
- quatre grilles dynamiques extraites à partir des informations volumiques produites par FDS : toxicité, température, feu et fumée,
- une grille dynamique, réactualisée à chaque pas de temps de simulation : la carte d'occupation de l'espace par les individus.

Notre contribution concerne la couche supérieure de la pyramide proposée par Funge et al. [FTT99]. Elle a pour ob-

jectif de modéliser le raisonnement de l'individu et la prise de décision qui en découle. Cette décision se matérialise par un déplacement souhaité, soit un point de l'espace que l'individu souhaite atteindre au pas de temps suivant de la simulation, compte tenu de la vitesse avec laquelle il est en mesure de se déplacer, des contraintes de déplacement et de son objectif d'évacuation du bâtiment. Notre modèle étant microscopique, autrement dit chaque individu raisonne au même instant de façon indépendante, rien ne garantit que les souhaits de déplacement des individus seront compatibles entre eux et en particulier qu'ils ne conduiront pas à des collisions entre individus. Il est donc nécessaire de valider ces déplacements afin de produire un ensemble de déplacements individuels compatibles entre eux et l'environnement, puisqu'il faut également éviter les collisions avec les obstacles tels que les murs. L'ensemble des déplacements souhaités est donc ensuite injecté dans le moteur physique Bullet**, qui renvoie un ensemble de déplacements valides. Puis le rendu réaliste de la simulation est effectué via la bibliothèque d'animation d'avatars Cal3D††.

4. Modèle comportemental et cognitif

La couche comportementale et cognitive constitue la principale contribution de cet article et est donc décrite en détails. Le modèle fonctionnel fait intervenir quatre modules cognitifs (voir figure 2) dont le but est de prendre la décision de déplacement en fonction de la perception subjective qu'a l'agent de son environnement. Chacun de ces modules va maintenant être décrit en détails.

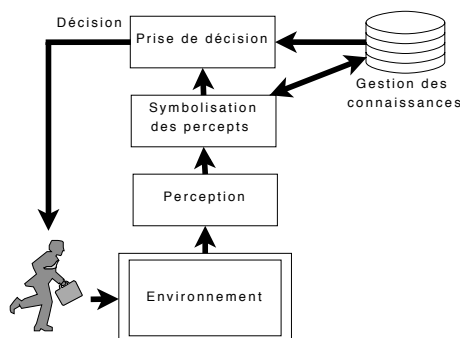


Figure 2: Modèle cognitif simulé

4.1. Perception

Ce module est chargé de collecter les informations en provenance de l'environnement. Les différents percepts captés par les agents sont le feu, les murs du bâtiment, les autres agents, les signaux logiques de communication, la

température, la fumée quantifiée par son indice de toxicité et les panneaux d'aide à l'évacuation. Parmi les 5 sens dont est doté l'homme, 3 sont utilisés et donc simulés :

- le toucher est simulé en sondant la cellule où se trouve l'agent,
- l'ouïe est simulée en détectant la présence de l'agent dans un cercle centré autour d'un autre agent, matérialisant la portée de la voix de cet agent,
- la vision est simulée par lancer de rayon, avec prise en compte de l'opacité selon la formule classique :

$$\alpha_{obs} = \int_0^L \alpha(s) ds \quad (1)$$

avec α_{obs} l'opacité cumulée le long du rayon, L la longueur du rayon et $\alpha(s)$ l'opacité locale à la distance s de l'origine du rayon. Les rayons sont arrêtés lorsque l'opacité cumulée atteint un certain seuil.

La table 2 donne la liste complète des informations perçues, leur mode de perception ainsi que la formule mathématique utilisée pour quantifier la donnée.

4.2. Symbolisation des percepts et gestion de la connaissance

Le module de symbolisation des percepts est chargé, pour chaque percept, de construire une représentation symbolique de la contrainte engendrée par ce percept. Les symboles seront ensuite manipulés par les algorithmes de simulation de la prise de décision.

Nous avons fait la distinction entre comportements navigationnels et comportements planifiés (voir section 2) et nous avons vu que les informations prises en compte par chaque type de comportement n'étaient pas forcément identiques.

Nous organisons la mémoire individuelle en 2 niveaux :

- La mémoire à court terme - MCT - stocke les symboles associés aux comportements navigationnels, autrement dit qui ne sont pas associées à la résolution d'un but. Exemple : force répulsive due à la présence du feu.
- La mémoire à long terme - MLT - est utilisée pour les symboles associés aux contraintes liées à la résolution d'un but. Exemple : densité de la foule.

Chaque type de comportement sera simulé par un algorithme spécifique qui travaillera avec un des 2 niveaux de la mémoire uniquement : la mémoire à court terme pour les comportements navigationnels et la mémoire à long terme pour les comportements planifiés. Les deux niveaux ne sont toutefois pas étanches et les symboles de la mémoire à court terme sont transférés dans la mémoire à long terme. Par exemple la présence du feu génère instantanément une force répulsive mais cette information sera également utilisée pour la planification.

L'information utilisée par un individu pour prendre une décision évolue au cours du temps. Elle est enrichie par les nouvelles connaissances acquises via la perception qu'il a de son environnement mais elle s'appauvrit également au cours du temps et ce pour deux raisons : l'oubli - involontaire - et la crédibilité accordée à l'information - volontaire.

** . <http://www.bulletphysics.com>

†† . <https://gna.org/projects/cal3d>

Nous ne simulons pas l'oubli mais gérons la fiabilité temporelle de l'information, dont le principe consiste à supprimer les informations lorsque ces dernières n'ont pas été validées par une observation effective plus ou moins récente, ce lap de temps étant fonction de la nature de l'information : on peut considérer qu'un mur aperçu une heure plus tôt a toutes les chances d'être encore à la même place, ce qui n'est pas le cas pour un groupe d'individus par exemple. Nous implémentons ce concept en attachant à chaque symbole un estampillage temporel correspondant à l'instant où cette information a été acquise. La fiabilité est alors définie sur l'intervalle [0,1] par la formule suivante :

$$fiab = 1 - \frac{(T_{cur} - T_{obs})}{T_{max}} \quad (2)$$

avec $fiab$ l'indice de fiabilité d'un symbole de contrainte dynamique, T_{obs} l'estampillage temporel de la dernière observation, T_{cur} l'instant courant et T_{max} le temps maximum de croyance à une observation dynamique, qui est propre à chaque agent.

Lorsque $fiab$ atteint 0, le symbole est considéré comme non fiable et est retiré de la mémoire.

La représentation informatique des symboles n'est pas la même selon le type de mémoire : représentation vectorielle - point dans l'espace et rayon - pour la mémoire à court terme et représentation discrète - grilles régulières - pour la mémoire à long terme. La raison de cette distinction tient à l'algorithmique utilisée : calcul d'une force répulsive dans le premier cas et algorithme du plus court chemin dans le second. Au niveau navigationnel chaque symbole vectoriel permettra de générer une force, déduite de la valeur du symbole et appliquée à l'individu. Le comportement navigationnel qui en découlera sera fonction de la somme des forces issues des différents symboles qui s'appliquent. Au niveau planifié, les grilles ont toutes même résolution et une combinaison linéaire de celles-ci produira une grille unique qui sera utilisée par l'algorithme de recherche de plus court chemin. Il eût été possible d'opter pour une autre structure, comme par exemple une surface triangulée, mais certaines contraintes sont dès le départ sous cette forme, telle la densité de la fumée par exemple, et les autres peuvent facilement être échantillonnées afin d'obtenir une représentation uniforme. Nous avons choisi une algorithmique compatible avec ce type de structure discrète que sont les grilles régulières, de la classe des Fast Marching Methods [Set99].

Pour le problème de l'évacuation, les principaux symboles associés aux contraintes sont :

- C_{Tox} : la toxicité perçue, issue du percept de fumée,
- C_{Temp} : la température perçue, issue du percept de température,
- C_{Mur} : les obstacles perçus, issus des percepts de feu et de murs,
- C_{Vital} : le respect des espaces vitaux des congénères, issu des percepts d'agents,
- $C_{Panneau}$: l'aide perçue, issue des percepts de panneaux d'aide à l'évacuation.

4.3. Prise de décision

L'architecture de comportement développée étant hybride planifiée/navigationnelle, les deux types de comportements y sont représentés.

4.3.1. Comportements navigationnels

Les comportements navigationnels utilisés sont des règles de répulsion/attraction associées à la contrainte de mouvement qu'ils représentent (voir table 2). Les symboles étant représentés sous forme vectorielle dans la mémoire à court terme, la force est évaluée en calculant la normale pondérée par une fonction de la distance : $\frac{i}{d^k}$.

4.3.2. Comportements planifiés

Contrairement aux précédents, ces comportements impliquent l'existence d'un but à atteindre et la décision qui en découle va avoir pour objectif de se rapprocher de ce but. Face à plusieurs buts possibles, l'agent va devoir effectuer un choix. Ce choix sera essentiellement fonction de deux paramètres : la qualité du but et l'effort nécessaire pour l'atteindre.

Par exemple imaginons que l'agent connaisse 2 issues possibles : une issue de secours lointaine et une porte plus proche mais incertaine car elle pourrait être fermée à clé. On voit bien que le choix n'est pas évident, sauf dans le cas où la seconde porte serait sur le chemin de la première évidemment, et que 2 individus confrontés à un tel dilemme pourraient faire des choix différents. Il est donc nécessaire de prendre en compte ces 2 paramètres afin de bien appréhender le problème.

Nous commençons par caractériser qualitativement les buts, puis nous expliquons comment nous mesurons l'effort nécessaire au déplacement et enfin le mécanisme de décision conduisant au choix effectué par l'agent.

Caractérisation des buts

Trois catégories de buts complémentaires peuvent être distingués :

1. Se rendre à une sortie connue et pouvant être atteinte. Le but est alors la sortie et on lui attribue la valeur 0 car son atteinte correspond à la satisfaction complète du problème puisque l'agent a évacué le bâtiment.
2. Suivre une personne qui déclare connaître une issue. Le but est alors de rallier l'endroit où se situe cette personne.
3. Rechercher les bords du bâtiment.

La numérotation implique une hiérarchisation qui doit amener l'agent à privilégier les buts de la catégorie 1, puis ceux de la catégorie 2 et enfin ceux de la catégorie 3. A chaque catégorie nous assignons une valeur numérique fonction de l'importance relative des buts de cette catégorie par rapport à ceux des autres catégories : 0 pour la catégorie 1, des valeurs strictement positives pour les autres, avec une valeur faible pour la catégorie 2 comparée à la catégorie 3. Ces valeurs seront injectées dans l'équation mathématique qui

intégrera également le coût du déplacement pour atteindre le but. Ainsi l'agent pourra choisir un but de la catégorie 2 même s'il existe un but dans la catégorie 1. Un individu peut par exemple, bien qu'ayant connaissance d'une issue de secours, préférer suivre un autre individu qui lui indiquerait connaître un raccourci.

En l'absence de sortie accessible connue, nous considérons qu'un individu va malgré tout raisonner afin de se diriger vers ce qui peut représenter une issue potentielle. Autrement dit, un individu non soumis à la panique ne fait jamais n'importe quoi : il raisonne afin de tenter d'évacuer le bâtiment. Par conséquent il y a toujours existence d'un but et donc planification, notre système étant par conséquent inductif.

Une issue d'un bâtiment est nécessairement une ouverture vers l'extérieur et donc, en l'absence de la connaissance précise d'une telle issue, toute cellule au bord du bâtiment est considérée comme une issue potentielle. En l'absence de but, nous aurions alors uniquement des comportements navigationnels et nous pourrions observer des phénomènes d'oscillation, tels qu'un individu oscillant entre deux murs. Or nous considérons que, dans une situation d'évacuation de bâtiment en feu par un individu non pris de panique, ce type de comportement n'est pas réaliste. Nous pensons qu'il ne peut survenir que comme la résultante d'un état de panique et nous l'avons donc uniquement autorisé dans cette situation.

Notons cependant qu'à tout moment au cours de ses déplacements, l'individu acquiert des informations via sa perception de l'environnement, ce qui l'amène à remettre en cause le but poursuivi, et à en changer lorsqu'un but de valeur supérieure lui apparaît. Un tel but peut être de même niveau - découverte d'une issue plus proche que celle qui était visée - ou d'un niveau supérieur - rencontre d'une personne ayant connaissance d'une issue par exemple. L'agent est ainsi en permanence engagé dans un processus exploratoire, au sens où il est à l'affût de tout but meilleur que celui qui est le sien actuellement. Cependant, dans la situation particulière où il n'a pas de but de niveau 1 ou 2, ce processus exploratoire prend alors tout son sens ou plus exactement le sens qu'on lui prête habituellement, à savoir la recherche d'une issue en l'absence de toute issue connue. A noter enfin qu'il n'y a aucun risque de comportement oscillatoire durant cette phase exploratoire puisque la connaissance liée aux obstacles n'est jamais remise en cause et donc lorsque l'agent se retrouve nez à nez avec un mur, il apprend et intègre désormais dans son comportement la présence d'un mur à cet endroit.

Coût des déplacements

Il ne peut y avoir de stratégie d'évacuation et donc de choix en l'absence d'un coût associé à l'atteinte d'un but. C'est ce qui fait choisir à un individu une sortie proche plutôt qu'une plus lointaine. Chaque contrainte va ainsi, suite au processus de symbolisation, être représentée par une grille dans la mémoire à long terme. Les différentes grilles vont ensuite être combinées, dans la mémoire à long terme de chaque agent, afin d'obtenir une grille unique qui représente l'ensemble

des contraintes de déplacement appliquées à l'agent. Les coefficients de cette formule peuvent être différents d'un agent à l'autre, introduisant une possibilité, parmi d'autres, d'individualisation des comportements.

L'expression générale de notre fonction de coût est :

$$C = \frac{Inconfort}{1 + Confort} \quad (3)$$

avec C la fonction de coût, $Inconfort$ une influence négative, représentant une répulsion locale, et $Confort$ une influence positive, représentant une attraction locale.

Cette formule implique que tout ce qui gêne le déplacement fait augmenter le coût de celui-ci alors que tout ce qui le facilite en diminue le coût.

Pour le problème de l'évacuation, nous définissons l'inconfort ainsi :

$$Inconfort_i = CDist + \sum_i fiab_i \times \alpha_{C_i} \times C_i \quad (4)$$

avec $CDist$ une contrainte constante symbolisant le coût de déplacement dans un espace non contraint, $fiab_i$ la fiabilité attribuée à l'observation du percept associé à la contrainte i par cet agent (voir équation 2), C_i la valeur associée à la contrainte i , normalisée sur l'intervalle $[0,1]$, et α_{C_i} le coefficient d'importance associé à la contrainte C_i pour cet agent. En l'absence de contrainte, ne subsiste que $CDist$, la distance devenant alors le seul critère pertinent.

Le confort est essentiellement constitué des aides à l'évacuation et est défini ainsi :

$$Confort_i = \sum_i \beta_{A_i} \times A_i \quad (5)$$

avec A_i la valeur associée à l'aide A_i , normalisée sur l'intervalle $[0,1]$, et α_{A_i} le coefficient d'importance associé à l'aide A_i pour cet agent.

Les jeux de coefficients α_{C_i} et α_{A_i} constituent l'archétype de comportement et caractérisent chaque individu. C'est en cela que notre modèle est microscopique et qu'il permet de générer une large gamme de comportements.

Prise de décision planifiée

Une fois les buts identifiés et caractérisés et la fonction de coût pour un agent calculée en toute cellule de la grille, la prise de décision se résume à trouver le plus court chemin menant l'agent au but le plus proche au sens de la fonction de coût et de la caractérisation de ce but.

La théorie des Level Sets [Set99] fournit une définition consistante du problème, qui a l'avantage de s'affranchir des contraintes de connexité, au sens topologique. Concrètement elle permet de calculer la distance de tout point à l'objectif le plus proche, qu'il y en ait un seul ou plusieurs, ce qui est adapté à notre problème puisque nous cherchons le chemin le plus proche à une issue du bâtiment, quelle qu'elle soit.

Partant de notre fonction de coût stockée sous forme de grille, nous utilisons une variante de la Fast Marching Method [Set99] - FMM - afin de trouver la solution au problème. Cette méthode simule la propagation d'un front équidistance, qui recouvre progressivement tout le domaine en

partant des objectifs, estampillant chaque cellule avec sa distance à l'objectif le plus proche. Le résultat est une carte des distances. L'intérêt est qu'une fois que ce calcul a été réalisé, la direction du plus court chemin à partir d'un point quelconque est fournie directement par le gradient de la carte de distances. En utilisant un structure de données appropriée, on parvient à obtenir une complexité de la FMM en $O(N \log(N))$, N étant le nombre de cellules de la grille. Nous détaillons les calculs mis en oeuvre à la section 5.2, en particulier l'implémentation du mécanisme de prise de décision sur GPU.

4.3.3. Implémentation du principe de rationalité limitée

Le modèle microscopique de sélection du comportement déclenché à un instant t s'effectue à travers deux phases successives :

- une phase ascendante capte les percepts, les symbolise et les stocke dans la mémoire,
- une phase descendante calcule le comportement planifié et les comportements navigationnels à partir des symboles stockés dans la mémoire.

Il est possible d'inhiber le niveau planifié afin de limiter la production de comportements au niveau navigationnel. L'individu continuera alors de naviguer correctement, en évitant les obstacles qui surviennent devant lui, mais sera incapable de raisonner et en particulier de replanifier son chemin. On crée ainsi un contexte de décision en accord avec les hypothèses de la Rationalité Limitée [Par05]. C'est ce mécanisme que nous avons retenu afin d'implémenter le comportement d'un individu en situation de panique, si important pour la compréhension des accidents en situation d'urgence. Nous proposons que la conséquence de l'état de panique soit une limitation des capacités cognitives de l'individu. Concrètement chaque individu est doté d'une jauge mesurant son degré de panique et lorsque le niveau de cette jauge atteint un certain seuil, l'individu est considéré comme sujet à la panique et le niveau cognitif/planifié est inhibé. Cette jauge est alimentée par différents événements survenant dans l'environnement de l'individu : par exemple si un individu découvre que l'issue qu'il espérait atteindre ne peut plus être atteinte, alors le niveau de sa jauge va augmenter.

5. Implémentation GPU

L'essentiel du coût en calcul de notre simulateur est dû à l'exécution de la FMM, aussi nous en proposons une implémentation sur GPU.

5.1. Processus de satisfaction de but

Le processus de satisfaction de but pour un agent s'effectue en plusieurs étapes :

1. Mise à jour des cellules buts G (processus de perception) : CPU,
2. Mise à jour de la grille de coût C (voir équation 3) : CPU,

3. Calcul de la carte des distances (voir 5.2) : GPU,
4. Détermination du gradient puis de la vitesse de déplacement de l'agent (voir équation 6) : GPU.

Ces différents traitements sont non seulement fortement parallélisables - traitement identique pour chaque cellule - mais doivent être effectués indépendamment pour chacun des agents.

5.2. Planification sur GPU

La Fast Iterative Method [JW07] est une variante de la FMM particulièrement adaptée aux architectures de type GPU. Elle consiste à chaque itération à réévaluer la distance de toutes les cellules au front, par opposition à la FMM qui limite cette opération aux cellules 4-voisines du front. En conséquence la FIM n'a pas besoin de gérer une liste des cellules voisines mais effectue beaucoup plus d'opérations que la FMM. Par contre, déployée sur une architecture GPU capable de réaliser l'opération en parallèle pour toutes les cellules en même temps, elle est par contre plus efficace puisque la complexité au pire est en $O(N)$ contre $O(N \log(N))$ pour la FMM, le surcoût pour la FMM étant dû à la gestion du front. L'algorithme 1 est celui de la FIM [MZ06]. D'un pas de temps au suivant durant la si-

Algorithm 1 Fast Iterative Method

Phi : champ de potentiel

Pour chaque cellule i **Faire** {Initialiser le champ Φ }

Si la cellule i appartient à un but **Alors**

$\Phi[i] \leftarrow 0$

Sinon

$\Phi[i] \leftarrow \infty$

Fin Si

Fin Pour

Répéter

$Stable \leftarrow Vrai$

Pour chaque cellule i **Faire**

$old \leftarrow \Phi[i]$

 Calculer $\Phi[i]$ {Réévaluation du potentiel}

Si $\Phi[i] < old$ **Alors**

$Stable \leftarrow Faux$

Sinon

$\Phi[i] \leftarrow old$

Fin Si

Fin Pour

Jusqu'à $Stable$

mulation, la connaissance sur laquelle raisonne un agent ne change pas beaucoup, sinon cela signifierait que le pas de temps est trop important et ne permet pas de capturer les évolutions avec suffisamment de précision. La conséquence est que la solution à la planification est elle aussi très ressemblante à celle calculée au pas de temps précédent. Afin d'exploiter cette cohérence temporelle, nous apportons deux modifications à l'algorithme de la FIM :

- on ne réinitialise plus le champ Φ à chaque pas de temps mais on repart avec la solution calculée au pas de temps précédent, ce qui permettra la plupart du temps de converger beaucoup plus rapidement,
- on modifie le test - ligne barrée dans l'algorithme - afin de valider systématiquement les changements de potentiel, y compris lorsque le potentiel augmente. Dans la version d'origine, le potentiel est systématiquement réinitialisé à l'infini et donc seules les diminutions de la distance aux objectifs sont validés. A partir du moment où nous repartons d'une solution existante, il est nécessaire d'envisager que la distance augmente. En effet, l'effort nécessaire pour atteindre un but peut augmenter, par exemple parce que l'agent vient de découvrir qu'un couloir qu'il comptait emprunter a été envahi par le feu. Tout changement de la distance est par conséquent validé.

L'algorithme 2 est notre algorithme de la FIM modifié pour les besoins de la simulation d'évacuation de bâtiment en feu.

Algorithm 2 Fast Iterative Method modifiée

Phi : champ de potentiel

Répéter

Stable \leftarrow *Vrai*

Pour chaque cellule **i Faire**

old \leftarrow $\Phi[i]$

Calculer $\Phi[i]$ {Réévaluation du potentiel}

Si $\Phi[i] \leftarrow old$ $\Phi[i] \neq old$ **Alors**

Stable \leftarrow *Faux*

Sinon

$\Phi[i] \leftarrow old$

Fin Si
Fin Pour

Jusqu'à *Stable*

5.3. Calcul du champ de forces

Le moteur physique Bullet^{††}, que nous utilisons pour la résolution des collisions, admet en entrée un ensemble de corps physiques et les forces qui s'appliquent sur eux, pour calculer un état d'équilibre du système garantissant l'absence de collisions. Nous devons donc calculer la force à appliquer à chaque individu et nous procédons comme suit. Une fois la carte des distances calculée, le calcul du gradient au centre des cellules est effectué par le GPU, ce qui évite de devoir rapatrier la texture en mémoire centrale.

Le gradient donne la direction du déplacement mais il nous faut de plus disposer de la vitesse de ce déplacement. Celle-ci serait donnée par la norme du gradient. Cependant, on peut admettre que dans une situation d'urgence, les individus adoptent systématiquement la vitesse maximale afin de

se mettre hors de danger. Les seules limitations à la vitesse sont d'ordre physique : vitesse de course, présence d'obstacles qui obligent à ralentir, inertie, visibilité, voire toxicité si on voulait modéliser le fait qu'un manque d'oxygène dans l'air affecte les capacités respiratoires et donc d'effort des individus. Nous calculons la vitesse comme une fonction des contraintes qui l'affectent :

$$V(x) = \prod_i (1 - \beta_{C_i} C_i(x)) \quad (6)$$

avec $V(x)$ la vitesse à adopter dans la cellule x , $C_i(x)$ la valeur associée à la contrainte i dans la cellule x , normalisée sur l'intervalle $[0,1]$ et β_{C_i} l'impact de la contrainte C_i sur la vitesse de déplacement.

La force résultante \vec{F}^{plan} est alors calculée ainsi :

$$\vec{F}^{plan} = - \frac{\nabla \Phi(x)}{\|\nabla \Phi(x)\|} V_x \quad (7)$$

6. Conclusion

Une architecture dédiée à la simulation microscopique du comportement d'individus en situation d'évacuation de bâtiment en feu a été présentée. Elle est fondée sur les deux principaux axiomes de théorie de la Rationalité Limitée proposée par H. Simon :

- informations disponibles limitées : gestion dynamique d'une mémoire individuelle à 2 niveaux : court terme et long terme,
- moyens cognitifs limités : contrôle du niveau de symbolisation de chaque contrainte.

L'architecture permet de décrire une mécanique décisionnelle humaine sur plusieurs niveaux d'abstraction et offre ainsi un moyen de contrôle de l'adaptation du raisonnement d'un individu confronté à un problème rationnel spatial et soumis à différentes contraintes de mouvement. Par rapport à la plupart des travaux existants, nous proposons un modèle très poussé de cette mécanique décisionnelle. En particulier, nous nous affranchissons des concepts d'omniscience et d'intention fanatique. De plus, dans le contexte de situations d'urgence, telles que l'évacuation de bâtiment en feu, ce modèle nous permet de simuler la panique des occupants, élément clé pour comprendre les comportements et apporter des réponses appropriées.

Enfin, cette architecture présente un fort potentiel de parallélisation qui a été mis en exergue à travers l'illustration d'un système de planification individuel sur GPU. Son application pour le problème de la simulation d'évacuation de bâtiment montre de très bonnes performances : 16 fps pour 100 agents sur GPU Nvidia GTX275. Les performances temps réel de notre solution ne sont pas à comparer avec celles des simulateurs de foule utilisés pour les applications de jeux vidéo car l'individualisation des comportements, fondement de notre approche, nous prive naturellement de la possibilité de factoriser les traitements, base de l'optimisation des cadences de rendu par les projets concurrents [TCP06].

††. <http://www.bulletphysics.com>

Lorsqu'il n'est plus possible de stocker la grille entière en mémoire GPU, nous basculons vers une architecture hybride CPU-GPU, la grille étant découpée en plusieurs sous-grilles chargées tour à tour sur le GPU, ce qui conduit à une grille à 2 niveaux : une sur-grille gérée par le CPU et des sous-grilles gérées par le GPU. Le rôle du GPU est d'appliquer le calcul de la FIM modifiée à la sous-grille que lui envoie le CPU, qui doit lui optimiser l'ordre dans lequel il envoie les sous-grilles au GPU car la convergence globale est très dépendante de cet ordre.

Notre implémentation du principe de rationalité limitée est assez basique puisqu'un individu est soit serein soit sous l'emprise de la panique alors qu'en réalité il y a probablement différents niveaux de panique. Il est ainsi des situations où ce modèle donnera des résultats improbables, comme par exemple le fait qu'un individu sous l'emprise de la panique et passant devant une porte ouverte sur l'extérieur ne prendra pas cette porte, parce que cela nécessiterait une replanification alors que le niveau planifié est inhibé. Il y a donc encore matière à réflexion autour de ce modèle.

Si notre modèle cognitif intègre la possibilité pour les individus d'utiliser des aides à l'évacuation, la signalisation notamment, nous n'avons pas pour l'instant considéré l'utilisation de dispositifs de lutte contre l'incendie, tels que les extincteurs. Plus tard, un lien avec les travaux de Paris et Donikian sur les BIIO pourra être fait dans cet objectif [PD09]. Une autre poursuite du développement de ce modèle pourrait consister en l'adjonction d'une couche logique évoluée et par suite d'une couche de collaboration entre individus permettant de modéliser l'entraide.

Références

- [AOM03] AMOR H. B., OBST O., MURRAY J. : Fast, neat and under control : Inverse steering behaviors for physical autonomous agents, 2003.
- [Fer97] FERBER J. : *Les systèmes multi-agents : vers une intelligence collective*. Dunod, décembre 1997.
- [FSJ01] FEDKIW R., STAM J., JENSEN H. : Visual simulation of smoke. *Computer Graphics (SIGGRAPH 2001)* (juillet 2001), 15–22.
- [FTT99] FUNGE J., TU X., TERZOPOULOS D. : Cognitive modeling : knowledge, reasoning and planning for intelligent characters. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 29–38.
- [HB08] HARABOR D., BOTEVA A. : Hierarchical path planning for multi-size agents in heterogeneous environments. In *IEEE Symposium on Computational Intelligence and Games* (Perth, WA, décembre 2008), pp. 258–265.
- [Hel07] HELIOVAARA S. : *Computational Models for Human Behavior in Fire Evacuations*. PhD thesis, Department of Engineering Physics and Mathematics, Helsinki University of Technology, 2007.
- [HLTC03] HEIGEAS L., LUCIANI A., THOLLOT J., CASTAGNÉ N. : A physically-based particle model of emergent crowd behaviors. In *Graphicon 2003 Conference Proceedings* (Moscou, Russie, Fédération De, 2003), p. non paginé.
- [HMF00] HELBING D., MOLNÁR P., FARKAS I., BOLAY K. : Self-organizing pedestrian movement. *Environment and Planning B : Planning and Design*. Vol. 28, Num. 3 (mai 2000), 361–383.
- [JW07] JEONG W. K., WHITAKER R. T. : *A Fast Iterative Method for a Class of Hamilton-Jacobi Equations on Parallel Systems*. Tech. rep., CiteSeerX - Scientific Literature Digital Library and Search Engine [http://citeseerx.ist.psu.edu/oai2] (United States), 2007.
- [KD05] KULIGOWSKI E. D., D. PEACOCK R. : Review of building evacuation models, juillet 2005.
- [LKL02] LITE D., KOENIG S., LIKHACHEV M. : D* lite, 2002.
- [MZ06] MANOLIOS P., ZHANG Y. : *Implementing survey propagation on graphics processing units*. Springer Verlag.
- [PAB07] PELECHANO N., ALLBECK J. M., BADLER N. I. : Controlling individual agents in high-density crowd simulation. In *SCA '07 : Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2007), Eurographics Association, pp. 99–108.
- [Pan06] PAN X. : *Computational Modeling of human and social behavior for emergency egress analysis*. Civil and environmental engineering dept., Civil and Environmental Engineering Dept, Stanford University, juin 2006.
- [Par05] PARTHENAY C. : Herbert simon : rationalité limitée, théorie des organisations et sciences de l'artificiel, 2005.
- [PBPD02] PROULX G., BEHAVIOUR U. H., PROULX G., D P. : Understanding human behaviour in stressful situations, 2002.
- [PD09] PARIS S., DONIKIAN S. : Activity-driven populace : a cognitive approach for crowd simulation. *IEEE Computer Graphics and Applications special issue Virtual Populace* (juillet 2009).
- [PJS06] PHILIPPSSEN R., JENSEN B., SIEGWART R. : *Autonomous Navigation in Dynamic Environments*. Springer Tracts on Advanced Robotics, 2006, ch. Towards Real-Time Sensor-Based Path Planning in Highly Dynamic Environments.
- [Rey06] REYNOLDS C. : Big fast crowds on ps3. In *Sandbox '06 : Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames* (New York, NY, USA, 2006), ACM, pp. 113–121.

- [Set99] SETHIAN J. A. : *Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1999.
- [SOB08] SHOPF J., OAT C., BARCZAK J. : *GPU Crowd Simulation*. GPU crowd simulation, AMD, Inc., 2008.
- [Spe96] SPEITEL L. C. : Fractional effective dose model for post-crash aircraft survivability. *Toxicology*. Vol. 115, Num. 1-3 (1996), 167 – 177. International Colloquium on Advances in Combustion Toxicology.
- [SV03] SETHIAN J. A., VLADIMIRSKY A. : Ordered upwind methods for static hamilton–jacobi equations : Theory and algorithms. *SIAM J. Numer. Anal.*. Vol. 41, Num. 1 (2003), 325–363.
- [TCP06] TREUILLE A., COOPER S., POPOVIĆ Z. : Continuum crowds. *ACM Trans. Graph.*. Vol. 25, Num. 3 (2006), 1160–1168.
- [Tha08] THALMANN D. : *Crowd Simulation*. Springer-Verlag London Limited, 2008.
- [VGL09] VEERASWAMY A., GALEA E., LAWRENCE P. : Implementation of cognitive mapping, spatial representation and wayfinding behaviours of people within evacuation modelling tools. In *Proceedings of the 4th International Symposium on Human Behaviour in Fire* (Robinson College, Cambridge, UK, juillet 2009), pp. 501–512.
- [WvdBM09] WILKIE D., VAN DEN BERG J., MANOCHA D. : Generalized velocity obstacles. In *IROS'09 : Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems* (Piscataway, NJ, USA, 2009), IEEE Press, pp. 5573–5578.