

# EvaPlanner: An Evacuation Planner with Social-based Flocking Kinetics

Cheng-Te Li

Graduate Institute of Networking and Multimedia,  
National Taiwan University, Taipei, Taiwan  
d98944005@csie.ntu.edu.tw

Shou-De Lin

Dept. of Computer Science & Information Engineering  
National Taiwan University, Taipei, Taiwan  
sdlin@csie.ntu.edu.tw

## ABSTRACT

This paper demonstrates a system that exploits graph mining, social network analysis, and agent-based crowd simulation techniques to investigate the evacuation dynamics during fire emergency. We create a novel evacuation planning system, *EvaPlanner*, to deal with three tasks. First, the system identifies the preferable locations to establish the exits to facilitate efficient evacuation from the dangerous areas. Second, it determines the most effective positions to place the emergency signs such that panic crowd can quickly find the exits. Third, it faithfully simulates the evacuation dynamics of crowd considering not only the individual movement kinetics but also the social connections between people. *EvaPlanner* provides a flexible experimental platform for investigating the evacuation dynamics under a variety of settings, and can further be utilized for animation and movie production. In addition, it can serve as a tool to assist architects address the safety concern during the planning phase. The demo system can be found in the link: <http://mslab.csie.ntu.edu.tw/evaplanner/>.

## Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—*Data mining*.

## General Terms

Algorithms, Performance, Design.

## Keywords

Social network, crowd simulation, evacuation planning.

## 1. INTRODUCTION

*Pedestrian evacuation and fire safety* is an important research area that studies the interactions between pedestrian movement and complex spatial environment under a variety of scenarios (e.g. terrorism, fire, or natural disasters), with the goal to implement a more effective evacuation plan to improve the ultimate survival rate. To perform such study, researchers have to consider the arrangement of objects in a given space, the deployment of exits and signs, and the identification of most direct routes toward the closest exits. Many practical issues, ranging from mental status, spatial cognition, panic factors, and group coordination are discussed from the perspectives of physics, physiology, psychology, and computer simulation (refer to the survey [11]). While most of existing works conduct qualitative studies to explore the evacuation dynamics, agent-based simulation is believed to be one of the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*KDD'12*, August 12 – 16, 2012, Beijing, China.

Copyright 2012 ACM 978-1-4503-1462-6 /12/08...\$15.00.

most effective approaches to explore the evacuation dynamics based on quantitative models.

Instead of relying completely on agent-based simulation, this paper integrates data mining techniques, in particular graph mining and social network models, with crowd simulation mechanism, to tackle the evacuation planning problem. We present a novel evacuation planning system, *EvaPlanner*, to explore and simulate the escaping dynamics of people in certain emergency situations. More specifically, we concentrate on answering three challenges. (a) Given any room structures and the deployment of interior objects and walls, where should the exits be setup to achieve better evacuation efficiency? (b) Where to put the signs pointing to the exits to optimize the survival rate during fire accident? (c) How to faithfully model, simulate, and visualize the crowd movement during evacuation for animation creation?

**(a) Automatic Exit Determination.** From the prospect of safety, it is critical to determine where to place the exits in a large room (e.g. exhibition or hall) during the blueprinting stage of the construction. Proper design of the location of the exits can improve the survival rate during fire. This paper proposes a novel idea to transform the exit-location determination task into an optimization problem in a graph and solve it with a graph-based random-walk mechanism.

**(b) Automatic Sign Placement.** Being able to place exit signs at the proper positions can also lead to more effective and efficient evacuation. In general, the signs have to be established in certain positions that are clearly visible, such that they are on the tracks of the shortest paths to the nearest exits. We transform the sign placement problem into a graph mining problem through exploiting graph-based traversing model, and then take advantage of the centrality measures designed for social networks as heuristics to identify the effective spots to put the signs.

**(c) Evacuation Simulation for Fire Accidents.** Here the goal is to simulate the realistic movement and evacuation traces of agents during fire emergency. Actually there have been proposals (e.g. [2][4][9]) exploiting techniques of crowd simulation to produce the escaping dynamics of agents in an indoor space. In these models, the persons are assumed to be independent agents and do not possess social relationships with each other. The key novelty of our simulation model is that it considers not only the kinetics of individual movement but also the social impact during evacuation. Naturally when experiencing an emergency such as a fire accident, people do not run away without attempting to rescue their family or friends. In this sense, people who are acquainted with each other tend to move and flock together during evacuation; therefore we believe a faithful simulation model should consider this social factor. Besides the basic movement kinetics proposed by Reynolds [10], we introduce a new steering force to model the acquaintance relationships between individuals. In the simulation, we first execute the exit-placement and sign-placement algorithms

to find the best locations to place exits and signs, and then perform crowd simulation with the devised social force to model the evacuation of humans during a fire accident.

## 2. SYSTEM FRAMEWORK

The *EvaPlanner* system framework is shown in Figure 1, which consists of three major components (drawn with dotted rectangles). The first component sets up the environment, which allows users to determine where to deploy the obstacles and agents in the environment. The second component is the core engine of the evacuation planner, which automatically determines the locations of *exits* and *signs* using graph mining techniques. The final component integrates the above modules to simulate the crowd movement under the *propagation of fire*. The main sub-component in the simulation is the *social-based flocking kinetics*, which aims to combine the flocking behavior and social acquaintance between agents into their collective movement.

Next, we will first describe how to intelligently determine the locations of exits and signs in Section 3, and the simulation module with social-based flocking kinetics in Section 4. We present the system demonstration in Section 5.

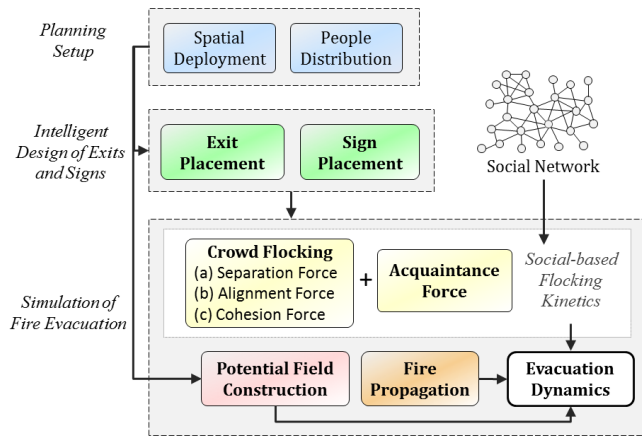


Figure 1: The system framework of *EvaPlanner*.

## 3. INTELLIGENT DESIGN OF EXIT & SIGN

Given an indoor space with scattered objects representing furniture and walls separating the rooms, our goal is to design an intelligent method to automatically determine the best positions on the border to implement the exits for smoother evacuation. Furthermore, once the locations of the exits are determined, we want to then determine the best spots to put the exit signs to optimize the total number of safely-evacuated persons. We devise graph mining approaches to fulfill these goals.

### 3.1 Exit Determination

For safety purpose, it is preferable to place exits that maximize the evacuation efficiency. To realize this idea, we propose a random-walk based mechanism for exit deployment. Our method consists of three steps. First, we transform the closed environment into a *navigation graph*. The navigation graph is devised to maintain the connectivity among spatial areas (represented by grids), where some grids are blank (i.e. indicates free-to-move spaces) and others are occupied by obstacles (i.e. impassable). For example, in Figure 4(a), the white grids are free areas while the shadowed grids are impassable areas. In the navigation graph, nodes represent the free grids and edges stand for the eight-directional neighboring relationship between grids (see Figure 4(b)). Second,

we apply the *Random Walk* (RW) algorithm [12] on the navigation graph. We randomly select some grids as the initialization nodes to perform RW. Third, as the RW process are executed up to  $N$  times ( $N=100$  in our system), we can derive the average reach probability of each grids. Those peripheral grids with highest probabilities are the preferable locations for exits as they are more likely to be reached by agents. In the demo system, we choose one exit for each peripheral side in the space.

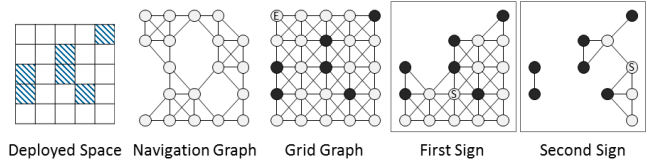


Figure 2: Transforming (a) the deployed space into (b) the navigation graph and (c) the visibility graph. The running example for (d) the first sign and (e) the second sign placed.

### 3.2 Sign Placement

The exit signs, which are set up to guide panic people towards the exits during emergency, play a key role in the evacuation process. Consequently, where to put the sign in the indoor environment can affect the evacuation efficiency. A preferable location to place the signs usually follows the following rules: (R1) it should be easily visible in the room, (R2) it indicates the shortest path to the nearest exits, and (R3) it should be placed at the prime locations where most people have to pass through. To incorporate these concepts, we propose to transform this problem into an optimization problem on a graph, and solve it using a greedy method with network analysis techniques.

We propose a three-step approach to greedily identify locations for placing signs pointing towards the exits. First, we transform the deployed space into a *grid graph*, which is a lattice representation of the space with the eight-directional connectivity. The grid graph aims to capture the spatial relationships between free spaces and obstacle spaces. For example, Figure 2(c) is the grid graph of Figure 2(a), in which white nodes are free grids while black nodes are the ones with obstacles. Given the derived exits, the second step aims to remove the free grids which can directly see the exits because they do not need to be guided by a sign. To do so, we first construct the shortest paths of each free grid to each exit. For each free grid, if any of its shortest paths contains no obstacle grid, then such grid (as well as its edges) is removed from the graph because we are sure there is a direct view from it to one of the exits. In other words, every free node that has at least one shortest path to the exit without being blocked will be removed from the grid graph because later we do not need to consider them for sign placement. After removing the visible grids, we then obtain a *visibility graph*. For example, assuming the only exit, marked as “E”, is located at the top-left position in a grid graph, as shown in Figure 2(c), the corresponding visibility graph is shown in Figure 2(d), where free nodes that are visible from the exit are removed. The third step is to determine where to put the sign based on the visibility graph. Here we compute the *closeness centrality* [13] and the *betweenness centrality* [13] values of each node in the visibility graph, and choose the grid with the highest average score of closeness and betweenness as the place for the first sign. Then we can go back to step-2, removing all the free-grids that have direct visibility to the first sign to obtain a new visibility graph as Figure 2(e), and then repeat step-3 to pick the place for the second sign. Such process continues until no free grids are left

in the visibility graph. For example, the first sign, marked as “S”, is placed at the center-bottom position in Figure 2(d), and the second sign is placed at the middle right of Figure 2(e). Note that choosing grids with the highest closeness scores usually satisfies R1 and R2, while choosing the grids with highest betweenness scores generally fit the criteria of R3.

#### 4. SIMULATION OF FIRE EVACUATION

In this section, we aim to integrate the previously discussed evacuation planning engine with an agent-based simulation framework to simulate how crowd moves during an emergency fire accident. The main novelty of this component lies in the introduction of a social-oriented moving kinetics to faithfully model the human’s behavior given social ties.

##### 4.1 Social-based Flocking Kinetics

We construct the social-based flocking kinetics of agents by devising the flocking steering forces and a social acquaintance force to jointly drive the movement of agents. First, we exploit Reynolds’ flocking model [10] as the basic movement guideline for agents. This model was proposed to model the flocking behavior of humans in a dynamic virtual environment. It consists of three steering rules, (a) *separation force* ( $f_s$ ) steers each agent to avoid crowding local flockmates, (b) *alignment force* ( $f_a$ ) steers each agent towards the average heading of local flockmates, and (c) *cohesion force* ( $f_c$ ) steers each agent to move toward the average position of local flockmates. Besides, each agent is an independent actor and has his own local perception of navigation (see Figure 3).

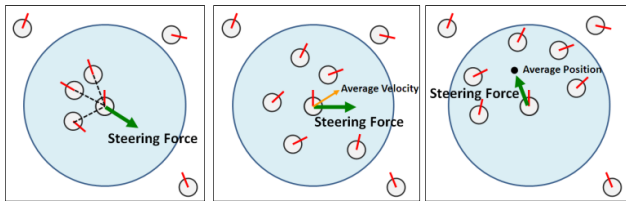


Figure 3: From left to right: illustrations of *separation*, *alignment*, and *cohesion* steering forces.

We also believe that besides these three factors, the movement of a human is also affected by his or her own social ties. That is, people who are acquainted with one another usually have higher chance to flock together in a particular spatial area [7]. Therefore, given a social network, a faithful simulation should provide a mechanism allowing people in the same community to have higher chance to flock together.

To emerge social-based flocking groups in the simulation, we introduce the *acquaintance force* ( $f_q$ ), into the flocking simulation. While the previous three forces are used to model the flocking behavior by gathering agents that are close enough in the simulating space, the integration of the acquaintance force is expected to produce the following effects: (a) determining the members of each flocking group, (b) steering agents whose corresponding nodes are close enough in the network to flock together, and (c) avoiding agents whose nodes are far apart to flock together. An illustration of the idea of acquaintance force is shown in Figure 4. For an agent/node  $u$ , the acquaintance force  $f_q(u)$  is computed by

$$\overline{f_q(u)} = \frac{1}{|R_u|} \sum_{v \in R_u} 1 - \frac{Length(u, v)}{\theta}$$

where  $R_u$  is the set of agents under the local perception of agent  $u$ ,  $Length(u, v)$  is the length of shortest path between the node  $u$  and  $v$  in the network, and  $\theta$  is a constant value that determines the shift border between the attraction and repulsion forces. That says, if  $Length(u, v) < \theta$ , the agent  $v$  will exert an attraction force on  $u$ . If  $Length(u, v) > \theta$ ,  $v$  will exert an repulsion force on  $u$ . And if  $Length(u, v) = \theta$ ,  $v$  has no effect on  $u$ . In this work,  $\theta$  is set to be 3. In the end, by integrating our acquaintance force with previous three steering forces, the eventual force  $\overline{f_q(u)}$  to guide agent  $u$  is  $\overline{f(u)} = w_s \cdot \overline{f_s(u)} + w_a \cdot \overline{f_a(u)} + w_c \cdot \overline{f_c(u)} + w_q \cdot \overline{f_q(u)}$ , where  $w_s, w_a, w_c, w_q$  are the weights of separation, alignment, cohesion, and acquaintance forces respectively. We set them to be  $(w_s, w_a, w_c, w_q) = (0.3, 0.3, 0.3, 0.1)$  in our system.

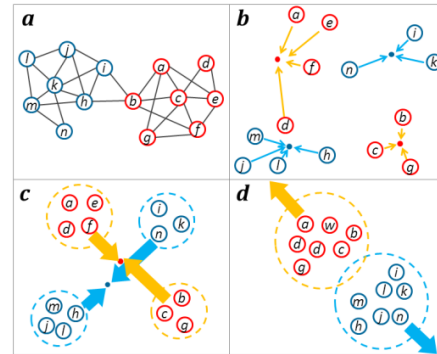


Figure 4: The effects of the acquaintance force. (a) An example social network containing two cohesive subgroups colored by blue and red. (b) The forces gradually gather local flocks together into some small groups whose members tend to be acquainted with each other. (c) The forces attract and merge some local flocks whose members are close enough in terms of graph distance. (d) For small groups whose members are farther apart in the network, repulsive forces will drive them away from one another.

##### 4.2 Potential Field Construction

Besides the mentioned four forces that guild the subtle movements of an agent in normal condition. We further propose to take advantage of the *potential field* [5] to guide the movements of agents in the deployed space during evacuation. The potential field is devised as a high-level force to drive the agents away from the obstacles or fire, and closer to the exits. The potential field we designed consists of two parts, the *guided attractive field*, which directs agents to the exits, and the *obstacle repulsive field*, which steers agents to bypass the obstacles. Figure 5 shows the idea of potential field, in which the exit has lowest potential while the obstacles possess highest potentials. Agents in the space tend to move from positions with higher potentials to positions with lower potentials. More details are described in the literature [3][5].

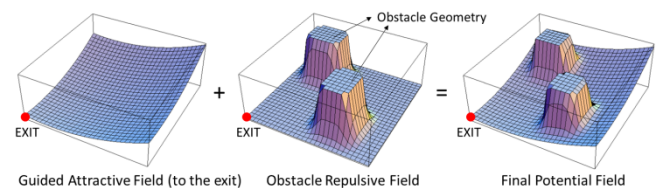


Figure 5: Combining (a) the *guided attractive field* and (b) the *obstacle repulsive field* to generate the final (c) *potential field*. Referred and modified from the material [3].

### 4.3 Fire Propagation and Evacuation

With the aforementioned mechanism, we can simulate the escaping behavior of agents during the propagation of fire. The fire evacuation simulation is continuously regulated by the following process, as described below. First, the fire events are initialized randomly in the closed space, and then gradually propagate to wider areas. Then the moving agents are notified about the fire events and starting to evacuate. During the evacuation process, family or friends would tend to move together given our flocking kinetics. As agents find the signs, they will learn the directions to the nearest exits. In the meantime, agents could be attacked by the fire and become unmovable (in the injured or dead status). This process continues until no more agents can move in the closed space. That says, agents either escaped or die in the fire. We report the number of evacuated and dead agents.

### 5. DEMONSTRATION

In addition to the abovementioned mechanism, *EvaPlanner* additionally provides the flexibility for users to setup the environment. To be more precise, given the initial blank/free space, *EvaPlanner* allow users to determine where to place obstacles or objects in the space. Such spatial deployment is fulfilled by allowing users to create some two-dimensional geometric obstacles and placing them onto the desired location in the free space. The remaining unoccupied area is considered as traversing spaces where people are allowed to pass through. Furthermore, users are allowed to determine the initial location of each agent. A random assignment of obstacles and agents in the space by our program can also be performed.

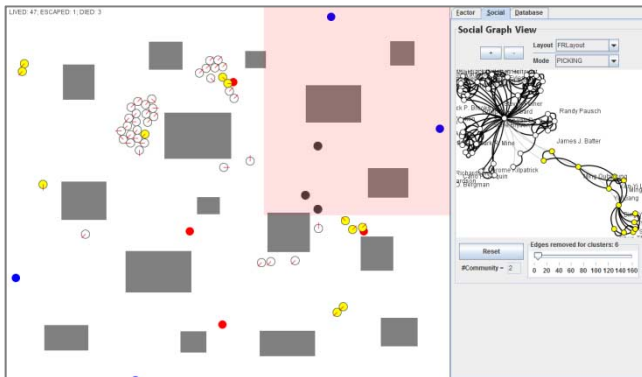


Figure 6: A system snapshot of *EvaPlanner*.

Figure 6 shows a snapshot of the evacuation process in our *EvaPlanner* system. The system consists of two parts, the simulation view on the left panel and the social network view on the right panel. In the simulation view, gray areas represent the obstacles, red circles indicate the emergency signs, and blue circles indicate the exits. The normal agents are the circles with red arrows revealing their current moving directions. A community detection method [8] is performed on the social network and agents in different community will be colored differently (e.g. white and yellow in Figure 6). During the simulation, the light-red area, which represents the fire coverage, propagates gradually. Agents touched by the fire will turn black and cannot move anymore, indicating injured or dead. We can observe that some agents moving in communities, such as the group at the top-left part of the graph. The survival statistics is also shown in the top-left cor-

ner as “LIVING: 47, ESCAPED: 1, DIED: 3.” Note that to understand the evacuation dynamics using our system, we create a demo video, which can be accessed via <http://mslab.csie.ntu.edu.tw/evaplanner/>.

### 6. CONCLUSION

While existing works on evacuation dynamics use mostly qualitative operations, this work presents a novel simulation-based evacuation planning system to model the evacuation dynamics. Systematically, we robustly integrate a series of important functions, especially on realistic social movement, intelligent design of exits and signs, and simulation of fire events. Technically, we exploit the techniques of social network analysis and graph mining [1][13] (from data mining), crowd simulation [10] (from computer animation), and motion planning [5] (from robotics) to develop the evacuation planner. Ongoing work focuses on three directions: designing different heterogeneous roles on agents (e.g. leader, trained, and untrained people), modeling sophisticated behaviors of agents (e.g. arsonist and fireman), and modeling the panic propagation [6] among agents.

### 7. REFERENCES

- [1] D. Chakrabarti and C. Faloutsos. Graph Mining: Laws, Generators, and Algorithms. *ACM Computing Survey*, 38(1), 2006.
- [2] S. Chenney. Flow Tiles. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 233-242, 2004.
- [3] Howie Choset. Robotic Motion Planning: Potential Functions. Lecture slides of motion planning, Robotics Institute 16-735.
- [4] D. Helbing, J. Farkas, and T. Vicsek. Simulating Dynamical Features of Escape Panic. *Nature*, 407, 487-490, 2000.
- [5] Y. K. Hwang and N. Ahuja. A Potential Field Approach to Path Planning. *IEEE Transactions on Robotics and Automation*, 8(1), 23-32, 1991.
- [6] D. Kempe, J. M. Kleinberg, and E. Tardos. Maximizing the Spread of Influence through a Social Network. *ACM SIGKDD*, 137-146, 2003.
- [7] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, 27:415-444, 2001.
- [8] M. E. J. Newman. Fast Algorithm for Detecting Community Structure in Networks. *Physical Review*, E 69, 066133, 2004.
- [9] N. Pelechano, J. M. Allbeck, and N. I. Badler. Controlling Individual Agents in High-Density Crowd Simulation. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 99-108, 2007.
- [10] C. W. Reynolds. Flocks, Herds and Schools. A Distributed Behavior Model. *ACM SIGGRAPH*, 25-34, 1987.
- [11] A. Schadschneider, W. Klingsch, H. Klüpfel, T. Kretz, C. Rogsch, and A. Seyfried. Evacuation Dynamics: Empirical Results, Modeling and Applications. *Encyclopedia of Complexity and Systems Science*, 3142-3176, 2009.
- [12] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast Random Walk with Restart and its Applications. *IEEE ICDM*, 613-622, 2006.
- [13] S. Wasserman and K. Faust. Social Network Analysis. *Cambridge University Press*, 1994.