Social Flocks: A Crowd Simulation Framework for Social Network Generation, Community Detection, and Collective Behavior Modeling

Cheng-Te Li Shou-De Lin
Graduate Institute of Networking and Multimedia,
National Taiwan University, Taipei, Taiwan
{d98944005, sdlin}@csie.ntu.edu.tw

ABSTRACT

This work combines the central ideas from two different areas, crowd simulation and social network analysis, to tackle some existing problems in both areas from a new angle. We present a novel spatio-temporal social crowd simulation framework, *Social Flocks*, to revisit three essential research problems, (a) generation of social networks, (b) community detection in social networks, (c) modeling collective social behaviors in crowd simulation. Our framework produces social networks that satisfy the properties of high clustering coefficient, low average path length, and power-law degree distribution. It can also be exploited as a novel dynamic model for community detection. Finally our framework can be used to produce real-life collective social behaviors over crowds, including community-guided flocking, leader following, and spatio-social information propagation. *Social Flocks* can serve as visualization of simulated crowds for domain experts to explore the dynamic effects of the spatial, temporal, and social factors on social networks. In addition, it provides an experimental platform of collective social behaviors for social gaming and movie animations. *Social Flocks* is at http://mslab.csie.ntu.edu.tw/socialflocks/ .

Categories and Subject Descriptors

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

General Terms

Algorithms, Performance, Design.

Keywords

Social Networks, Crowd Simulation, Flocking, Network Generation, Community Detection, Collective Social Behaviors.

1. INTRODUCTION

It is generally believed that human beings who live and interact in certain geographical area tend to form a social group. In this paper, we present a novel spatio-social simulation framework, *Social Flocks*, to model and exploit such anthropological natures by exploiting the technique of crowd simulation, which is to reveal collective behaviors by simulating the movement process of a number of agents. The central idea of *Social Flocks* is to assume each node as an agent that moves in the space guided by several different kinds of forces. This paper shows that *Social Flocks* can be exploited to solve three major problems in social network analysis and crowd simulation. First, it can be exploited to create networks that satisfy real-world property such as the small-world phenomenon and scale-free properties. Second, given a social network, *Social Flocks* can be utilized to identify network communities. Third, it can be treated as a simulation system that produces three kinds of social-based collective behaviors. The demonstration page of *Social Flocks* is available at http://mslab.csie.ntu.edu.tw/socialflocks/.

Social Network Generation. Social network generation models aim at producing artificial social networks satisfying some well-known properties that have been discovered in real-world social networks [13]. Three of the most essential properties are (a) high *clustering coefficient CC* (nodes are densely-connected to their neighborhood), (b) low *average path length APL* (all pairs of nodes are connected via short paths on average), and (c) *power-law degree distribution*. Watts and Strogatz [18] propose the *random rewiring* model to generate the *small-world* networks with high clustering coefficient. Barabasi and Albert [3] propose the *preferential attachment* mechanism to generate the *scale-free* networks that satisfy both (b) and (c). More advanced generative methods [1][4] have been proposed to model a series of sophisticated network properties as well. However, most of existing generation models ignores the fact that human societies are not formed by random rewiring or preferential attachment. In reality, ancient people migrated to the same geographical regions and gradually interact with each other to form societies. Therefore, in this work, rather than resorting to graph theory or topological composition methods, we exploit the technique of crowd simulation to develop a *CrowdNetGen* component for the generation of realistic social networks. *CrowdNetGen* integrates both spatial and temporal factors when forming a society. In *CrowdNetGen*, we propose two agent-based flocking mechanisms, the *touch* model and the *neighborhood-density* model, to generate social networks by gradually linking nodes that are in contact together. We also demonstrate that our approach is able to generate networks with high CC, low APL, and power-law degree distribution. The generated property values are similar to many empirical studies on real-world complex networks [13]. Though some studies [2][6][7] use the agent-based approach as spatial clues to generate networks, they do not investigate whether the structural properties satisfy those of the real-world.

Finding Communities. Community detection is a well-studied problem in social network mining. Generally it is tackled by first devising an objective function that captures the concept of the community structure (i.e., nodes within a cluster are tightly connected while nodes between communities are loosely connected). Though many methods have been proposed and compared (see Leskovec's review [11]) for community detection, most of them are topology-based approaches that overlook an important fact. Realistically, people form groups not due to the desires to optimize certain objective function, rather they form communities because they interact or contact with each other more frequently [12]. Based on the above insight, we present a new crowd simulation approach, termed *Crowdstering*, to identify communities in a social network.

KDD'11, August 21–24, 2011, San Diego, California, USA.

Simulating Collective Behaviors. Different from *CrowdNetGen* and *Crowdstering* that bring the idea of crowd simulation into a social network mining problem, here we exploit the opposite by bringing the social factors into a crowd simulation framework. Several models have been proposed to simulate crowds to produce collective behaviors, including social force [8], cellular automata [5], and rule-based model [16]. Others use physiological (e.g. locomotion, energy level) and psychological (e.g. impatience, personality attributes) traits of agents to trigger heterogeneous behaviors [15]. However, existing approaches do not consider the social interactions among the agents, thus they are unable to produce social-dependent scenarios. In the third part of this work, we propose to leverage the underlying social network, which captures social relationships among the agents, for crowd simulation. We propose a social network-based crowd simulation, *SocioCrowd*. *SocioCrowd* can simulate three social-based collective crowd behaviors, including *community-guided flocking*, *leader following*, and *spatio-social information propagation*. These real-world social behaviors cannot be correctly modeled using existing methods. The three social-based behaviors can be utilized to create intelligent characters in online gaming and visual effects for the movie industry.

2. *SOCIAL FLOCKS* FRAMEWORK

We first present the *Social Flocks* framework as shown in Figure 1. *Social Flocks* takes advantage of Reynolds' *flocking simulation* model [16] as the backbone, in which we associate each node in a social network with a moving agent to perform the following three tasks. The first part is *CrowdNetGen*. We propose two network generation methods, *touch* and *neighborhood-density* models, to produce networks with properties of high CC, low APL, and power-law degree distribution. Second, given a social network, we propose a new spatio-temporal simulation approach for community detection, *Crowdstering*, by introducing the *acquaintance force* derived from the network into the system. The third part is *SocioCrowd*. By considering the network structure as the social contexts among the agents, we simulate three social-based collective crowd behaviors, including community-guided flocking, leader following, and spatio-social information propagation.

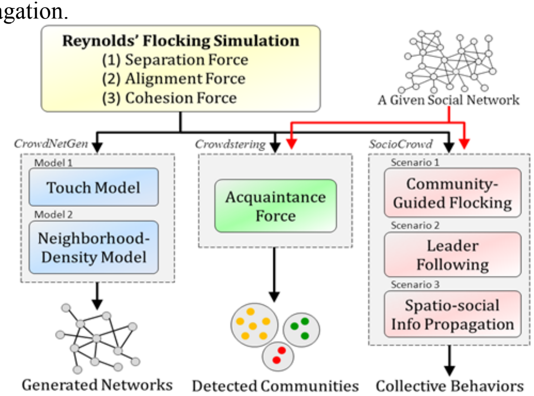

Figure 1: The *Social Flocks* Framework.

Here we briefly introduce Reynolds' flocking model [16]. It is proposed to capture the flocking behavior among artificial agents 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 to navigate.

3. SOCIAL NETWORK GENERATION

We exploit a completely different strategy in generating real-world social networks using crowd simulation. The proposed generative model considers each node as an agent. Then as the flocking simulation proceeds, we gradually add edges between nodes based on one of the following models[1].

Touch Model. This model aims to produce a network that reflects the way people in the pre-telecommunication era form groups by physically meeting each other in space. In the touch model, an edge is added to connect agent/node u and v only when u and v have a physical touch during the simulating process. Figure 2 shows that as #round increases from 0 to 1550, the network quickly gathers edges and both CC and APL drastically increase. When the #round reaches 600 to 1000, as highlighted by the orange rectangle, the generated networks possess the small-world properties of high CC≈0.45 and small-world APL≈6. Unfortunately, the touch model does not quite produce the scale-free property. The power-law exponent α is about 1.5, which is smaller than that of many real social networks (α≈2).

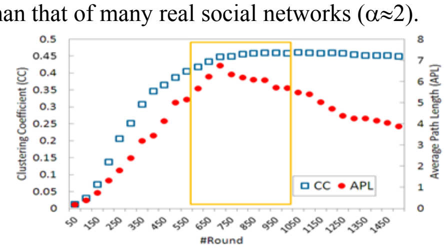

Figure 2: CC and APL under the *Touch* Model.

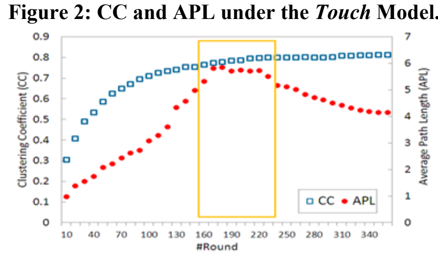

Figure 3: CC and APL under the *neighborhood-density* model.

Neighborhood-Density Model. To satisfy the three real-world properties in the generated networks, we devise the *neighborhood-density* model. The basic intuition is that an agent has higher likelihood to develop connections with others when there are more agents around, and furthermore it is more likely to develop relationship with the leader of a group. For an agent/node v, we define its *neighborhood-density* k_v as the number of neighboring agents within its local perception range ε (set to be 20 pixels here). During the flocking simulation, for each agent/node v, if k_v is larger than a predefined density threshold (set to be 5 in this work), the system adds an edge to connect v to a node u with the highest k_u value in v's local perception (i.e., such node is considered as the leader it perceives. Figure 3 presents the values of CC and APL during the simulation. We can see that as the rounds of simulation increases from 0 to 150, the network quickly gathers edges and both CC and APL increase drastically. The small-world properties emerge with high CC≈0.75 and low APL≈6 between round 150 and 200, as highlighted by the orange rectangle. As #round further increases, more edges are introduced to connect nodes from different flocks, and the APL gradually decreases. CC remains stable because edges are added for all nodes in a neighborhood at the same time (i.e., forming triangles in the network). Note that

[1] In this work, (f_s, f_a, f_c) = (0.5, 0.8, 0.5) for the *touch* model, and (f_s, f_a, f_c) = (0.7, 0.8, 0.5) for the *neighborhood-density* model.

comparing with the touch model, the neighborhood-density model takes fewer rounds to produce high CC and low APL. It is because the touch situations are relatively less likely to happen. In addition, the generated networks under the neighborhood-density model follow the power-law degree distribution. And the power-law exponent α at #round=170 is 1.92. In brief, the neighborhood-density model can produce networks that satisfy the three properties of real-world networks, i.e., high CC, low APL, and power-law degree distribution ($\alpha \approx 2$ in average).

4. FINDING NETWORK COMMUNITIES

In anthropology and evolutionary sciences, the geographic homophily plays a significant role on the formation of different kinds of human societies [12]. People who acquainted with each other usually live and flock in a particular spatial area and have higher potential to interact with one another. We consider such spatio-temporal homophily factor to develop a novel approach, *Crowdstering*, for finding communities in a network. Specifically, given a social network, our goal is to generate flocking groups in a natural way, with each flocking group corresponds to a detected community in the network. The natural way means that except for the steering forces that contributes to the spatio-temporal flocking behavior, our *Crowdstering* is parameter-free with respect to the prior knowledge about the communities, such as the number of communities.

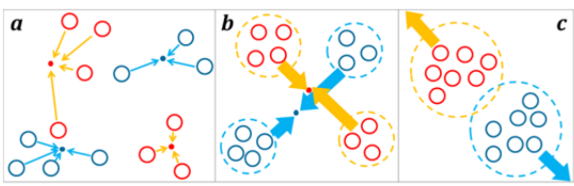

Figure 4: Expected effects of integrating the acquaintance force. (a) Gathering local flocks whose members tend to be acquainted with each other. (b) Attracting and merging local flocks gradually during the simulation. (c) For those far from acquaintances, the repulsions drive them away from one another.

To emerge flocking groups as communities in the simulating space, we introduce the fourth factor, the *acquaintance force* (f_q), into the flocking simulation. While the original three forces are used to produce the flocking behavior by gathering agents that are close enough in the simulating space, the integration of the acquaintance force is expected to emerge the 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) steering agents whose nodes are far apart in the network to avoid flocking 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

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

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 θ 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 to u. If $Length(u,v) > \theta$, v will exert an repulsion force to u. And if $Length(u,v) = \theta$, v has no effect on u. In this work, θ is set to be 3.

Here we use an example to show the effectiveness of *Crowdstering*. Given a network with two different sets of detected communities, the simulated outcomes are shown in Figure 5. For Figure 5(a), there are two detected communities, colored yellow and white, and the simulated outcome of our *Crowdstering* is nearly perfect as it emerges two flocking groups. For Figure 5(b), there are three detected communities, colored green, yellow, and white. We find that the previous white community is split into the current white and green communities, while the yellow community remains the same. Though the white and green communities flock together, the acquaintance force is still able to steer agents of each community to be closer in the network and towards two small cohesive flocking subgroups. In addition, we can find some agents who do not form a flock. They could be outliers that are loosely connected to any communities.

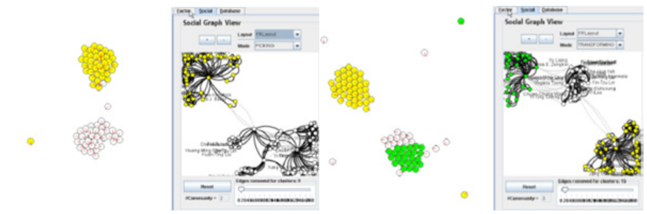

Figure 5: *Crowdstering* for a network with (a) two communities, and (b) three communities.

5. COLLECTIVE SOCIAL BEHAVIORS

Existing crowd simulation studies usually regard each simulation agent as independent (i.e., no social connection between each other). Although general crowd simulation methods can produce the realistic fine-grained human actions, they do not consider any real-life social interactions among the agents. This shortcoming prevents the creation of social-dependent crowd behaviors. Here we aim to leverage the underlying social network, which captures social relationships among agents, to produce collective social behaviors over crowds. The *SocioCrowd* component is developed to simulate three collective behaviors, including community-guided flocking, leader following, and spatio-social information propagation. We collect co-authorships from DBLP to construct the underlying network. An example social subgraph is shown in Figure 6(a).

5.1 Community-Guided Flocking

In the real life, it is assumed that people who are in the same community tend to interact with each other more frequently. We believe it is important that a simulation framework can reproduce such behavior. Therefore, we aim to simulate such common social scenario given a social network with detected communities. Note that in the demo we use the Fast Newman algorithm [14] to find communities community detection. For agents in diverse communities, we use different colors on both network view and 2D view. To integrate communities within the simulation, we design two probabilities, p_c and q_c, to control the possibility of an agent being attracted by others of the same and different community respectively. The simulating results are shown in Figure 6(c), where p_c=1 and q_c=0. It can be observed that agents belonging to the same communities tend to flock. For Figure 6(b), where p_c=0.5 and q_c=1, agents belonging to different communities have higher chance to flock.

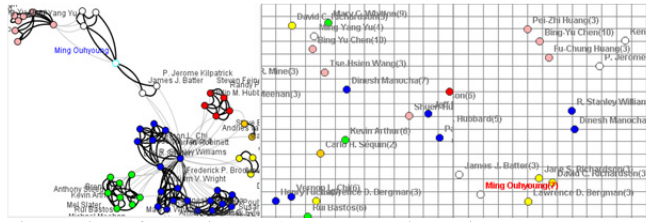

Figure 6: Left to right: (a) Network view with seven detected communities. (b) Simulating behaviors without community effect. (c) Collective behaviors of community-guided flocking.

5.2 Leader Following

A society usually has few leaders who are followed by other ordinary citizens. *SocioCrowd* simulates such social scenario by identifying some central individuals as the leaders in a social network. Three centrality measures (i.e., degree, closeness, and eigenvector) [17] are provided to find central individuals from different viewpoints. During simulation, we design a following probability p_l which allows an individual to follow a leader given the leader appears in the individual's local perception. A higher probability implies the leader attracts more followers. Figure 7 shows the result of leader following behavior, where $p_l = 0.7$.

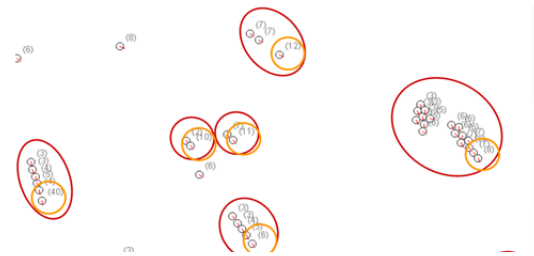

Figure 7: The emerging collective crowd behaviors of leading person following. Each red circled group is a flock and the orange circled ones are the leaders in flocking groups.

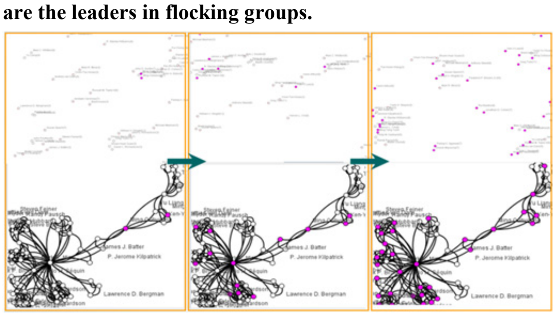

Figure 8: Certain information (purple), is propagated from few to most of the agents shown from left to right. The top is the simulating view while the bottom is the network view.

5.3 Spatio-Social Information Propagation

We simulate how information is propagated among agents. In conventional crowd simulation, agents spread messages only to others who are close to them in space. In real world, however, people do not necessarily communicate with spatially adjacent individuals but rather with socially adjacent ones. With underlying social network, individuals are capable of interacting with their social friends. *SocioCrowd* combines spatial and social clues to perform such information spreading. We adopt the linear threshold model [9] as the spreading strategy, with which an agent is influenced if the summation of the influence levels of its spatial and social neighbors is above a given threshold. A snapshot of influenced agents drawn in red cylinders is shown in Figure 8.

6. CONCLUSIONS

Generally, researchers regard crowd simulation as a sub-area in computer animation, which is not very relevant to the study of social network analysis. This paper, however, discovers a connection between crowd simulation and social network analysis and utilizes such connection to enhance performance on both sides. We first exploit crowd simulation framework to solve two important social network problems, namely generation of a social network that fits the real-world property, and community detection. Comparing with exiting models on network generation and community detection, our solution is able to reflect how ancient people form societies and visualize the dynamics of the generation process. Therefore users can further observe how the social links or communities are generated and utilize dynamic information to pursue further analysis. On the contrary, we also show social networks can benefit the area of crowd simulation. Previous models treat agents as individuals and do not consider the implicit social networks behind them. We have shown that by considering the forces derived from the social connection, it is possible to produce several collective social behaviors, including community-guided flocking, leader following, and spatio-social information propagation. Ongoing work will concentrate on modeling the evolving network properties, such as densification power law and shrinking diameter [10], and producing advanced collective social behaviors, such as riot and evacuation.

7. ACKNOWLEDGMENTS

This work was supported by National Science Council, National Taiwan University and Intel Corporation under Grants NSC99-2911-I-002-201, 99R70600, and 10R70500.

8. REFERENCES

[1] L. Akoglu, and C. Faloutsos. RTG: A Recursive Realistic Graph Generator using Random Typing. *PKDD* 2009.

[2] M. Aoyagi, and A. Namatame. Synchronization in Mobile Agents and Effects of Network Topology. *Agent-Based Social Systems*, 6(3) 31-42, 2009.

[3] A. L. Barabasi, and R. Albert. Emergence of Scaling in Random Networks. *Science* 286(5439), 509-512, 1999.

[4] D. Chakrabarti, and C. Faloutsos. Graph Mining: Laws, Generators, and Algorithms. *ACM Computing Survey*, 38(1), 2006.

[5] S. Chenney. Flow Tiles. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 233-242, 2004.

[6] B. Edmonds. How Are Physical and Social Spaces Related? – Cognitive Agents as the Necessary "Glue", *Agent-based Computational Modeling*, 5:195-214, 2006.

[7] L. Hamill, and N. Gilbert. Social Circles: A Simple Structure for Agent-Based Social Network Models. *Journal of Artificial Societies and Social Simulation*, 12(2), 3, 2009.

[8] D. Helbing, J. Farkas, and T. Vicsek. Simulating Dynamical Features of Escape Panic. *Nature*, 407, 487-490, 2000.

[9] D. Kempe, J. M. Kleinberg, and E. Tardos. Maximizing the Spread of Influence through a Social Network. *ACM SIGKDD*, 137-146, 2003.

[10] J. Leskovec, J. M. Kleinberg, and C. Faloutsos. Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations. *ACM SIGKDD*, 177-187, 2005.

[11] J. Leskovec, K. Lang, and M. Mahoney. Empirical Comparison of Algorithms for Network Community Detection. *ACM WWW*, 631-640, 2010.

[12] 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.

[13] M. E. J. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45(2), 2003.

[14] M. E. J. Newman. Fast Algorithm for Detecting Community Structure in Networks. *Physical Review*, E 69, 066133, 2004.

[15] 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.

[16] C. W. Reynolds. Flocks, Herds and Schools. A Distributed Behavior Model. *ACM SIGGRAPH*, 25-34, 1987.

[17] S. Wasserman, and K. Faust. Social Network Analysis. *Cambridge University Press*, 1994.

[18] D. J. Watts, and S. H. Strogatz. Collective Dynamics of 'Small-world' Networks. *Nature*, 393:440-442, 1998.