

Crowd Distribution and Location Preference

Weizi Li
George Mason University
wlia@gmu.edu

Zichao Di
George Mason University
zdi@gmu.edu

Jan M. Allbeck
George Mason University
jallbeck@gmu.edu
<http://cs.gmu.edu/~jallbeck/>

Abstract

Most crowd simulators focus on navigation and agents flow. In this paper, we present another perspective which concentrates on the overall distribution of virtual agents and uses psychological preferences for choosing goal locations. Both observation and published theory indicate that most people prefer to maintain their personal space as event spaces increase in density, particularly when they have no previous relationship to other individuals. The geometric structure that naturally forms could be highly approximated by a Voronoi tessellation. Our method allows users to specify sub-regions of an environment and tag the regions with information (e.g. permitted densities and features). A Centroidal Voronoi Tessellation (CVT) is then automatically constructed over the entire virtual world. The centers of mass of each resulting cell are taken as potential agent goal locations. Individual virtual agents then have the ability to choose their preferred goal locations based on their own characteristics (e.g. personality traits, needs, and interests), the CVT, and semantic features of the sub-regions. This method results in more meaningful crowd simulations with minimal additional user effort.

Keywords: CVT, Virtual Humans, Functional Populace

1 Introduction

Crowd simulation is an important research topic in computer graphics and virtual reality. It has great potential and attracts more and more attention. Most crowd simulators focus on navigation and the flow of agents. However, a large part of our daily life consists of more static, contextual activities rather than constantly traveling. For example, sitting through class, watching shows, and attending a variety of events. In addition, when participating in an event, we have preferences for locating ourselves within the venue. These preferences might stem from personality traits, needs, interests, environmental features, or other characteristics. For instance, at a concert some may prefer front row seats, while others prefer to be immersed in the crowd. In another example, conference participants may base their next goal location on their interests (to pick sections to attend) or on their needs (to get food, water, or social interaction).

In this paper, we present an agent-based approach, addressing crowd simulations from another perspective: population distribution. Instead of simulating agents traversing an environment to a randomly chosen place, we use characteristics of the agents and environment to determine goals for each agent. We explore the impact of psychological factors on the virtual agent goal location choosing. From the agent perspective we incorporate various psychologi-

cal variants such as personality traits, needs, and interests. From the environment perspective we adapt both spatial information and semantics.

First we use spatial information and user specified sub-region densities to construct a Centroidal Voronoi Tessellation (CVT) of a 2D floor plan of the environment. The sub-regions may be based on rooms or may just correspond to regions with special features or meanings. The calculated centers of mass of each cell of the CVT are considered potential goal locations. After that, agent characteristics are mapped to environment features, providing agents with the ability to generate their own preferred destination. While uniform density functions are normally applied when constructing CVTs, our method permits users to specify sub-regions with varying densities. This allows some attractions and areas to be more popular than others.

CVT's are featured Voronoi diagrams in which each generating point is also the center of mass of each cell. According to [1, 2], when creatures try to establish their territories in an area the resulting polygonal boundaries could be highly approximated by a Voronoi tessellation. Because of this feature, many researchers have been adapted Voronoi diagrams in simulating virtual agents walk around each other while maintain their personal space or obtain maximum clearance [3, 4, 5]. And this intuitively lead us to use the CVT's as the basis of our problem. Of course, human beings tend to be much more complex creatures. The population distributions are not always even. In fact some factors, such as human relationships, will modify the formed structure. We will discuss several issues of this kind in later sections.

In summation, this work's contributions to agent-based crowd simulations will include:

- Adapting CVT to determine agent goal locations, leading to simulations that are more reasonable than current randomly distributed goals, and lead to more consistent agent behaviors.
- Incorporating environment semantics and features with agent characteristics allowing agents to demonstrate preferences.

2 Related Work

There have been several approaches taken when trying to simulate crowds. First, much effort has been devoted for improving navigation and flow, including global path finding and local collision avoidance, for instance [6, 7, 8, 9]. Other researchers, in order to achieve more believable and reasonable agent interactions, employ data-driven approaches. Using video clips, researchers either construct individual or group behavior models or use the clips as an evaluation mechanism for their own behavioral models [10, 11]. Few of these publications address how agent goal locations are determined. For evacuation scenarios certainly *Exits* are needed. For others, it seems random locations or even more often a few predetermined sink locations are used.

Other work uses the concept of patches to add behavioral variations to the crowds [12, 13]. The general technique is to define several regions in the virtual world and then associate these regions with certain behaviors and regulations. Through entering different regions, agents interact with objects and other agents in certain patterns.

Group formation is another research focus [14, 15, 16]. Here the focus is also on navigation, but includes how to maintain the formations when environmental constraints are encountered. Often agents are given a predetermined position in the formation and equipped with simple repulsion forces to maintain the formation structure. Certainly, formations are not a natural distribution for many situations.

Most of the work described so far included few or no social psychological factors. Pelechano et al included roles and other psychological factors, such as panic, to vary agents behaviors in evacuation scenarios [17]. Shao and Terzopoulos describe a virtual train station where each agent role is associated with hand crafted action selection mechanisms [18]. In [19], the authors also simulate social roles and include role switching based on various socio-psychological factors. For extensive review of crowd simulation, we refer readers to [20, 21].

The concept of Centroidal Voronoi Tessellation (CVT) has been widely applied in computer graphics, image and data analysis and many re-

search areas [22, 23]. Particularly, A.Sud et al [3] is using first and second order Voronoi diagrams to create paths for virtual agents in facilitating route planning and proximity computation. Champagne and Tang compute 2D Voronoi diagrams on environment maps to navigate groups of agents through the virtual world [4]. In [5] CVT-based paths are formed for robots to follow in order to achieve maximal clearance.

The coverage problem is similar to how CVT is used in the method that we present. For instance, [24] addresses polygonal regions with multiple holes when attempting to deploy multiple robots to cover a space. [25] focuses on optimal sensor location finding and feedback control.

Other relevant work explores the spatial and semantic features of virtual environments. [26] incorporates both geometric and semantic information in order to find collision-free paths for robots. [27] builds a topological semantic location model, particularly focusing on terms such like *exit* and *path*. [28] simulates crowds on an informed environment by building an ontology that names each part of a virtual city.

Our method includes social psychological factors and addresses determination of goals positions to yield more reasonable crowd simulations. Furthermore, it allows users to specify environment semantics, alter the density within a space and modify the position preference, allowing scenarios to be more contextual.

3 Spatial Perspective

In this section, we will first give a brief introduction to CVT and the algorithm we adapted to construct it, and then formulate our problem.

3.1 Centroidal Voronoi Tessellation

A Voronoi diagram is defined as a map from the set of N -dimensional vectors in the domain $\Omega \subset \mathbb{R}^N$ into a finite set of vectors $\{\mathbf{z}_i\}_{i=1}^k$ called generators. Given a distance function $d(z, w)$, the Voronoi set V_i is the set of all elements belonging to Ω that are closer to z_i than to any of the other elements z_j , which is

$$V_i = \{w \in \Omega \mid d(w, z_i) < d(w, z_j), j = 1, \dots, k, i \neq j\}$$

For a given density function ρ defined on Ω , we may define the centroids, or mass centers, of regions $\{V_i\}_{i=1}^k$ by

$$\mathbf{z}_i^* = \frac{\int_{V_i} \mathbf{y} \rho(\mathbf{y}) d\mathbf{y}}{\int_{V_i} \rho(\mathbf{y}) d\mathbf{y}}.$$

A CVT is then a tessellation for which the generators of the Voronoi diagram coincide with the centroids of their respective Voronoi regions, in other words, $\mathbf{z}_i = \mathbf{z}_i^*$ for all i .

Given a set of points $\{\mathbf{z}_i\}_{i=1}^k$ and a tessellation $\{V_i\}_{i=1}^k$ of the domain, we may define the *energy function* for the pair $(\{\mathbf{z}_i\}_{i=1}^k, \{V_i\}_{i=1}^k)$ as

$$\mathcal{G}(\{\mathbf{z}_i\}_{i=1}^k) = \sum_{i=1}^k \int_{V_i} \rho(\mathbf{y}) |\mathbf{y} - \mathbf{z}_i|^2 d\mathbf{y}.$$

where $\rho(\mathbf{y})$ is the population density and $|\cdot|$ is Euclidean distance. The minimizer of \mathcal{G} forms a CVT which illustrates the optimization property of the CVT [22].

3.2 Truncated-Newton Algorithm

The most traditional and classical method for constructing CVTs is the algorithm developed by Lloyd in the 1960s [29] which represents a fixed-point type iterative mechanism. However, in our application, we chose Truncated-Newton algorithm (TN) to construct the CVT which is designed to solve large-scaled optimization problems and achieve high accuracy.

Here, we provide a short explanation of the Truncated-Newton algorithm [30]. To optimize a problem of the form

$$\min_{\mathbf{z}} f(\mathbf{z})$$

at the j -th iteration a search direction p is computed as an approximate solution to the Newton equations

$$\nabla^2 f(\mathbf{z}_j) p = -\nabla f(\mathbf{z}_j)$$

where \mathbf{z}_j is the current approximation to the solution of the optimization problem. The search direction p is computed using the linear conjugate-gradient algorithm (CG). The necessary Hessian-vector products are estimated using finite differencing. The TN algorithm only requires that values of $f(\mathbf{z})$ and $\nabla f(\mathbf{z})$ are computed. TN has low storage requirements, has

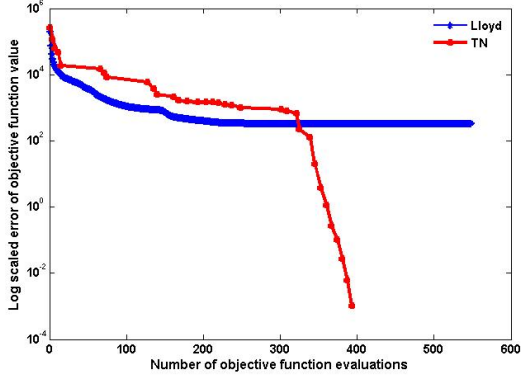


Figure 1: Convergence comparison between TN and Lloyd which is the usual benchmark used for CVT construction.

low computational costs per iteration, and hence is suitable for solving large optimization problems [31]. For procedure of applying TN on CVT’s construction, we refer interested readers to [32].

In Figure 1, we compare the convergence performance of TN with Lloyd which is the usual benchmark applied for CVT construction on square domains with uniform density. The experiment is performed on 100 generators. As x-axis represents, we measure the computational cost of TN by counting the number of function evaluation of \mathcal{G} , which estimates the dominant cost of using TN. For Lloyd, we count the number of equivalent centroid evaluations. y-axis is showing the log scaled error of optimal objective function corresponding to the needed computational cost for both TN and Lloyd as shown in x-axis. As we can see, Although Lloyd performs well at the first few iterations, TN is continually converging to optimal solution, while the converging of Lloyd is much slower and never be able to reach the same accuracy as TN. Overall, we can conclude that TN attains a higher order accuracy than Lloyd.

3.3 Potential Location Finding

As discussed in previous sections, CVT is an ideal structure in providing basis of our problem and the centers of mass are taken as potential agent goal locations. In addition, for this work, we make the following assumptions:

- Each agent’s personal space is represented

by a cell in the CVT.

- Relationships between agents are currently not represented or used in calculations.
- While agent characteristics and preferences are considered, more rich social interactions are not.

With these assumptions, it is not difficult to see that the centers of mass calculated from the CVT make good potential locations for virtual agents in generating plausible distributions. In addition, since our virtual world is built on a flat terrain, we can easily project an entire scene composition onto a 2D plane and then construct a CVT on it.

And as we informed earlier, users can choose densities for each sub-region. They can also specify their own sub-regions that may not coincide with permanent environment features. Our method permits five specific density scales: 0.2, 0.4, 0.6, 0.8 and 1.0. When all the densities have been assigned, a user can further refine the number of estimate generators in the scene. In other words, they can determine how many potential locations for virtual agents should exist in the environment. After these parameters are all given, a certain number of generators will be picked for each sub-region according to its size and density, and the actual CVT construction process will start. An example is shown in Figure 2.

4 Semantic Perspective

In this section, we will discuss both agent characteristics and environment features and most importantly how these two components are combined to enable our virtual agents to have preferred goal locations.

4.1 Agent Characteristics

Humans are very complex creatures with many elements influencing their decision-making. While our method is not specific to any particular set of characteristics, we have focused on personality traits, needs, and interests. For personality traits, we exploit one of the more popular models, the Five-factor or OCEAN model

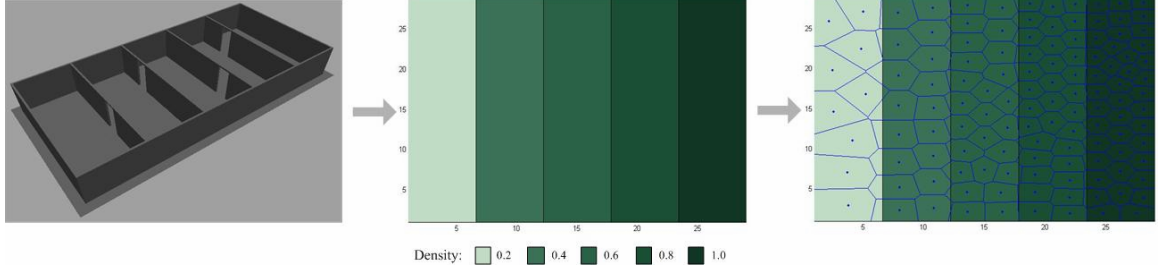


Figure 2: The left image shows a virtual environment in the 3D space which includes 5 sub-regions. The middle image presents the 2D layout of the virtual scene with assigned density information. The right picture provides the final CVT construction result. From left to right, the potential goal locations defined in each sub-region are: 9, 17, 25, 33 and 41.

[33]. The five factors are: *Openness*, *Conscientiousness*, *Extroversion*, *Agreeableness*, and *Neuroticism*.

Though the goals and applications differ, we have adopted an implementation of personality traits similar to [34]. An agent’s personality P is a five-dimension vector, and each is represented by a personality factor Ψ_i . The distribution of the personality factors in a population of agents is formulated by a Gaussian distribution function with mean μ_i and standard deviation σ_i :

$$P = \langle \Psi_O, \Psi_C, \Psi_E, \Psi_A, \Psi_N \rangle$$

$$\Psi_i = N(\mu_i, \sigma_i^2), \text{ for } i \in O, C, E, A, N$$

$$\text{where } \mu \in [-10, 10], \sigma \in [-2, 2]$$

Ψ as a bipolar, could be either positive or negative values. For instance, a positive for *Extroversion*, $E+$, means that the individual has an extroverted character; whereas a negative value means that the agent has an introverted personality.

In addition, we have chosen Maslow’s Hierarchy of Needs [35] as a foundation for representing virtual human needs. Conceptually, there is a reservoir associated with each need for each agent. The initial level of each reservoir is set randomly. After a certain amount of time has elapsed, each reservoir will decrease by a specified decay rate. When the level of a reservoir is lower than a predefined threshold, the agent will go to a place which has the corresponding functionality to fulfill the need. For example, if an agent is hungry, then he will find a food court. Once the agent has fulfilled his needs, the level

of reservoir will increase according to a fulfillment rate.

Another agent characteristic represented is interests. Each agent may have multiple interests. Thus far, we have allowed each agent to have five keywords associated with five themes of interest. Each keyword is paired with a weight (from 1 to 5), indicating to what degree the agent favors this topic. For example in a conference simulation, an agent’s interest might correspond to secession topics: $I = \{ \langle \text{virtual agents}, 5 \rangle, \langle \text{entertainment}, 4 \rangle, \langle \text{tutoring}, 3 \rangle \dots \}$.

In summation, the agent characteristics: $Agent_{char} = \langle P, N, I \rangle$ where $P = \langle \Psi_O, \Psi_C, \Psi_E, \Psi_A, \Psi_N \rangle$ are the personality traits, $N = \langle N_{food}, N_{water}, N_{excretion} \rangle$ are the needs, and I is a tuple representing agent interests where $I_j = \{ \langle \text{interest}_j, \text{weight}_j \rangle \}$, $j \in [1, 5]$.

4.2 Environment Features

When considering environment semantics, normally researchers would label each object and region in the virtual world and build a hierarchy to organize them. However, we find this insufficient. In this work, we also permit the tagging of sub-regions with additional information (i.e. functionality, size and density). As previously mentioned, we have five specific density scales that can be assigned to different areas. The size of each region can easily be calculated from the geometry or user defined areas. We include these features because we believe that they can impact the preferences of the agents. For example, agents with introverted personalities are less

likely to go to a very crowded and small area. Finally, functionality indicates the purpose of the space and links to agent needs and interests. For example, restrooms fulfill a certain need, as do cafeterias, and conference meeting rooms might be labeled with topics being presented.

In conclusion, an environment consists of features: $Env_{feature} = \langle D, S, F \rangle$ where D is the region density, S is the region size and F is the region functionality which is a tuple where $F_k = \{\langle function_k, weight_k \rangle\}$, $k \in [1, 5]$.

4.3 Feature Mapping and Decision Making

Naturally, agent characteristics might be in conflict when trying to determine a goal location. Priorities are used to mitigate these conflicts. In our method, each characteristic is assigned a priority value from one to the number of characteristics, where one is assumed to be the highest priority. For example, *needs* get priority level one while the *interests* and *personality* have level two and level three. This means that agent will first try to fulfill his needs if they are below the threshold and will ignore other matches from interests and personality. Secondly, if all needs are currently satisfied, then the agent will search for the most interesting place. If more than one place has the highest interest, then the agent will choose a place according to his personality traits. If none of the agent's characteristics can be used to determine a single goal location, then a location will be chosen randomly. An overview of mapping is shown in Figure 3 and a detailed description of each match is given below.

First of all, *needs* obtain priority level one, meaning if some of them require satisfying, the agent will preempt the current plan and go to the corresponding location to fulfill them. The mapping between this agent characteristic and the environment feature is straightforward, *needs* to *functionality*. For example, if an agent feels hungry, he will find a place that provides the food.

Secondly, *interests* have priority level two and since it is highly correlated with region functionality, we tie them together too. We simply match interests with functionality, taking into account the interest weights. For exam-

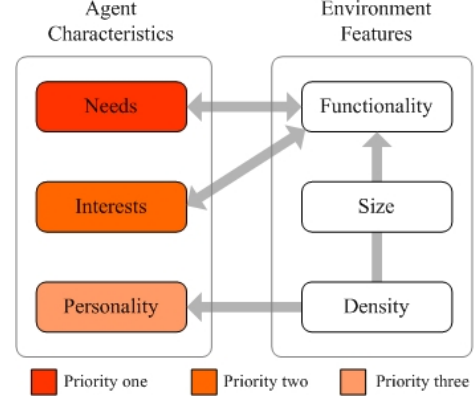


Figure 3: Mapping between agent characteristics and environment features.

Table 1: Mapping between personality traits and environment functionality, size and density.

Environment Features	Personality Traits
Functionality	O, C, E, A, N
Size	E, N
Density	O, E, A, N

ple, two regions might have functionality $F_1 = \{\langle virtual\ agents, 5 \rangle, \langle entertainment, 3 \rangle \dots\}$ and $F_2 = \{\langle robotics, 5 \rangle, \langle entertainment, 4 \rangle \dots\}$ and an agent interests $I = \{\langle virtual\ agents, 5 \rangle, \langle robotics, 4 \rangle, \langle entertainment, 2 \rangle \dots\}$. Assuming no matches among the unspecified elements, the first region would score: $5 \times 5 + 3 \times 2 = 31$ and the second region: $5 \times 4 + 4 \times 2 = 28$. As the first region received higher score, it would be chosen as the goal region.

Table 1 shows a mapping from personality traits to environment functionality, size and population density. Note that for simplicity only the personality dimension is represented in the table, not the valence of each trait. For example, *Extroversion* is positively correlated with region size and density, however, negatively with certain region functions such as *Lounge*.

The following details how agent region selection is calculated based on personality traits:

Functionality: remember for each region there exist five keywords with weights representing its five functions. So for i th agent, the functionality score of this region is:

$$S_{func} = \frac{\sum_{k=1}^5 \{weight_k \times \frac{\Psi_i^O + \Psi_i^C + \Psi_i^E + \Psi_i^A + \Psi_i^N}{0.2}\}}{0.2}$$

Size: we first compute the ratio between the size of each region and entire environment and then according to the results give each region a size scale ω ranging from 1 to 5. Then for i th agent, the size score is:

$$S_{size} = \omega \times \{(\Psi_i^E + \Psi_i^N) \times 0.5\}$$

Density: recall that we have five specific density scales (i.e. 0.2, 0.4, 0.6, 0.8 and 1.0) here we magnify them by 5 times to meet the computational scale which gives us a new density δ ranging from 1 to 5. Then for i th agent, the density score is:

$$S_{den} = \delta \times \{(\Psi_i^O + \Psi_i^E + \Psi_i^A + \Psi_i^N) \times 0.25\}$$

Now, after S_{func} , S_{size} and S_{den} has been computed, we obtain the overall score with α as weights:

$$S_{overall} = \alpha_{func} \times S_{func} + \alpha_{size} \times S_{size} + \alpha_{den} \times S_{den}$$

where $\alpha_{func}, \alpha_{size}, \alpha_{den} \in (0, 1)$ and $\alpha_{func} + \alpha_{size} + \alpha_{den} = 1$

The region with the highest $S_{overall}$ is chosen as the preferred region given the agent's personality.

5 Results

For our simulation, we are using an underlying crowd simulator [7] which is based on an automatically generated cell and portal graph structure and a hybrid rule-base/social forces model. The agents will start from locations such as *Entrances* and then walk to their goals.

The first example scenario entails simulating a conference scenario. As for assigning densities in this scenario, we assume that an event planner would have prior knowledge of various sessions (e.g. how much interest there might be for a session) and of the room capacities. Accordingly, the planner would specify the maximum number of potential spots in each room. The overview of this scenario is shown in Figure 4(a). The scene contains 13 rooms. The assigned density information and room numbers are providing in Figure 4(b). The total number of generators, or in other words total quantity of

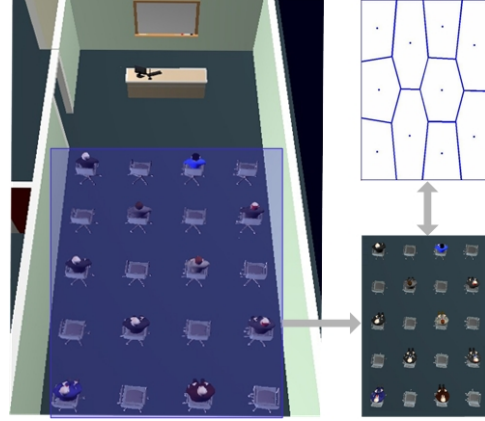


Figure 5: A classroom scenario with 10 agents. The agents choose a seat by finding the closest one to their CVT centroids.

potential locations in this example is 134 and the resulting CVT is showing in Figure 4(c).

Agents personality traits and need levels are set randomly as the simulation begins. The keywords of interests for each agent are chosen from following items: *virtual agents, robotics, innovative applications, main, entertainment, tutoring, social, lounge* and *none*. The final distribution of goal locations of 100 agents is shown in Figure 4(d).

Figure 5 shows another example. In this classroom scene, our model is extended to take into account the impact of chairs or other furniture on the agent distribution. This requires only one additional step. Each agent chooses the closest one regarding its CVT centroid.

In addition, we measured the average spacing between neighboring agents for *Room_12* both with and without the CVT. As we can see from Figure 6, with CVT the average spacing tends to be much more stable and maintain a reasonable level. Here we chose stable average distance as a metric, because given that agents have no social relationships (as we assumed before, which is a simplification), it's more natural that one maintain a relative equal distance from each other rather than get very close to one agent while staying far away from another.

6 Conclusion and Future Work

In this work, we aim to generate more believable crowd distributions and provide virtual agents

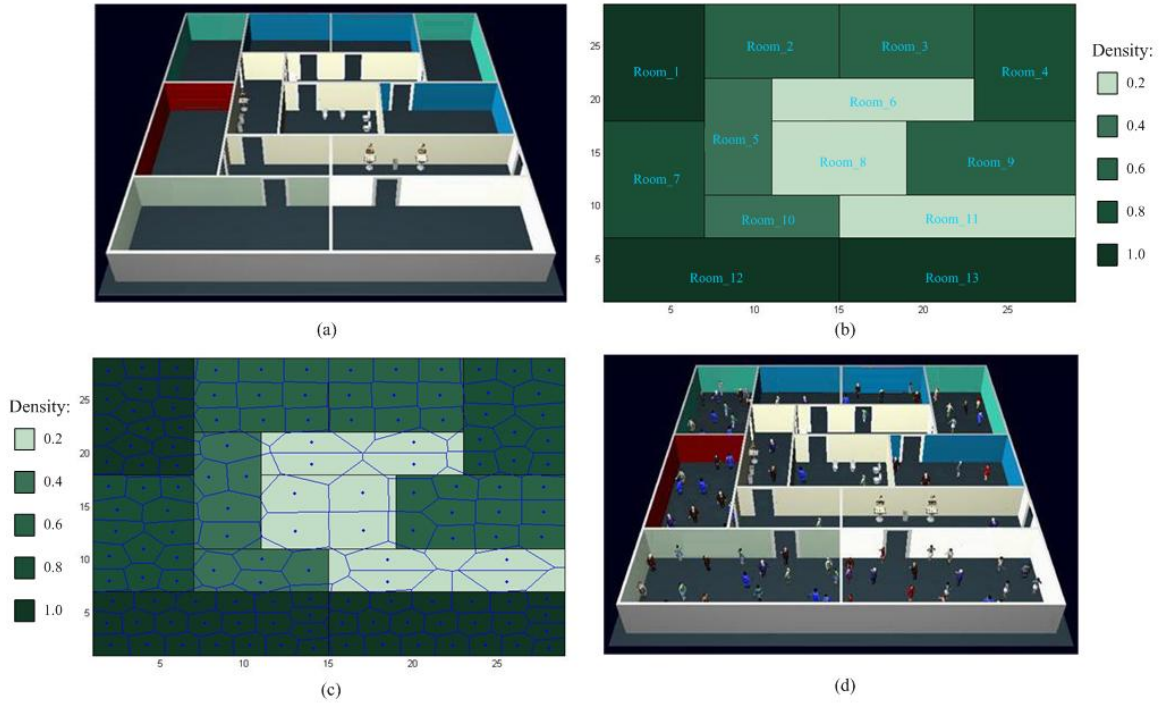


Figure 4: (a) The virtual conference scenario. (b) Detailed density information and room numbers. (c) Resulting CVT structure with 134 generators; From room 1 to 13, the number of locations are: 16, 9, 9, 13, 5, 4, 13, 4, 11, 4, 4, 21, 21. (d) Final distribution of 100 agents.

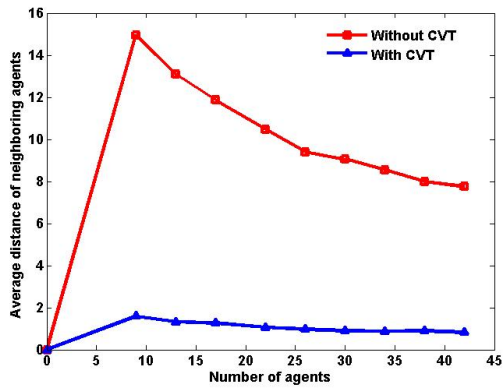


Figure 6: Average spacing between neighboring agents in *Room_12*

with the ability to choose their preferred goal locations. First, we adopted CVT to assist in finding an even distribution. The 2D layout of the virtual environment and user specified region densities are used to build a CVT. The centers of mass of each cell in the CVT are then considered as potential goal locations. Next, we simulate agent preferences by mapping agent characteristics (e.g. personality traits, needs, and interests) to environment features (i.e. functionality, size, and density). Agents then choose from potential goal locations based on this mapping.

Certainly this work could be extended to include more socio-psychological traits, such as roles, cultural variations and interpersonal relationships. Likewise other environment features could be specified and mapped to the agent characteristics.

Acknowledgements

Partial support for this effort is gratefully acknowledged from the U.S. Army SUBTLE MURI W911NF-07-1-0216. We also appreciate donations from Autodesk.

References

- [1] H. Honda. Description of cellular patterns by dirichlet domains: The two-dimensional case. *Journal of Theoretical Biology*, 72(3):523–530, June 1978.
- [2] A. Suzuki and M. Iri. Approximation of a tessellation of the plane by a voronoi diagram. *J. OPER. RES. SOC. JAPAN*, 29(1):69–97, 1986.
- [3] A. Sud, E. Andersen, S. Curtis, M. C. Lin, and D. Manocha. Real-time path planning in dynamic virtual environments using multiagent navigation graphs. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):526–538, May - June 2008.
- [4] J. Champagne and W. Tang. Real-time simulation of crowds using voronoi diagrams. In *EG UK Theory and Practice of Computer Graphics*, pages 195–201, 2005.
- [5] L. Zheng, Y. Choi, X. Liu, and W. Wang. CVT-based 2d motion planning with maximal clearance. In *Proc. of the 2011 IEEE International Conference on Robotics and Automation (ICRA 2011)*, pages 2281–2287. IEEE, May 2011.
- [6] A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. In *Proc. of the 2006 ACM SIGGRAPH Conference*, pages 1160–1168. ACM, August 2006.
- [7] N. Pelechano, J. M. Allbeck, and N. I. Badler. Controlling individual agents in high-density crowd simulation. In *Proc. of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2007)*, pages 99–108. ACM, August 2007.
- [8] R. Narain, A. Golas, S. Curtis, and M. C. Lin. Aggregate dynamics for dense crowd simulation. In *Proc. of the 2009 ACM SIGGRAPH Asia Conference*, pages 122:1–122:8. ACM, December 2009.
- [9] J. Ondřej, J. Pettré, A. Olivier, and S. Donikian. A synthetic-vision based steering approach for crowd simulation. In *Proc. of the 2010 ACM SIGGRAPH Conference*, pages 123:1–123:9. ACM, August 2010.
- [10] K. H. Lee, M. G. Choi, Q. Hong, and J. Lee. Group behavior from video: A data-driven approach to crowd simulation. In *Proc. of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2007)*, pages 109–118. ACM, August 2007.
- [11] J. Pettré, J. Ondřej, A. Olivier, A. Cretual, and S. Donikian. Experiment-based modeling, simulation and validation of interactions between virtual walkers. In *Proc. of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2009)*, pages 189–198. ACM, August 2009.
- [12] H. P. H. Shum, T. Komura, and S. Yamazaki. Simulating interactions of avatars in high dimensional state space. In *Proc. of the 2008 Symposium on Interactive 3D Graphics and Games (I3D 2008)*, pages 131–138. ACM, February 2008.
- [13] B. Yersin, J. Maim, J. Pettré, and D. Thalmann. Crowd patches: Populating large-scale virtual environments for real-time applications. In *Proc. of the 2009 Symposium on Interactive 3D Graphics and Games (I3D 2009)*, pages 207–214. ACM, February 2009.
- [14] M. Sung, L. Kovar, and M. Gleicher. Fast and accurate goal-directed motion synthesis for crowds. In *Proc. of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2005)*, pages 291–300. ACM, July 2005.
- [15] T. Kwon, K. H. Lee, J. Lee, and S. Takahashi. Group motion editing. In *Proc. of the 2008 ACM SIGGRAPH Conference*, pages 1–8. ACM, August 2008.
- [16] Q. Gu and Z. Deng. Formation sketching: An approach to stylize groups in crowd simulation. In *Proc. of the Graphics Interface 2011 (GI 2011)*, pages 1–8, May 2011.

- [17] N. Pelechano, K. O'Brien, B. Silverman, and N. I. Badler. Crowd simulation in incorporating agent psychological models, roles and communication. In *First International Workshop on Crowd Simulation (V-CROWDS 2005)*, pages 21–30, November 2005.
- [18] W. Shao and D. Terzopoulos. Autonomous pedestrians. In *Proc. of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2005)*, pages 19–28. ACM, July 2005.
- [19] W. Li and J. M. Allbeck. Populations with purpose. In *Proc. of the Fourth International Conference on Motion in Games (MIG 2011)*, pages 132–143. Springer, November 2011.
- [20] N. Pelechano, J. M. Allbeck, and N. I. Badler. *Virtual Crowds: Methods, Simulation, and Control*. Morgan & Claypool Publishers, San Francisco, 2008.
- [21] D. Thalmann. *Crowd Simulation*. John Wiley & Sons Inc, 2007.
- [22] Q. Du, V. Faber, and M. Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.
- [23] Q. Du, M. Gunzburger, and L. Ju. Advances in studies and applications of centroidal voronoi tessellations. *Numer. Math. Theor. Meth. Appl.*, 3(2):119–142, May 2010.
- [24] K. J. Obermeyer, A. Ganguli, and F. Bullo. Multi-agent deployment for visibility coverage in polygonal environments with holes. *International Journal on Robust and Nonlinear Control*, 21(12):1467–1492, August 2011.
- [25] Y. Chen, Z. Wang, and J. Liang. Optimal dynamic actuator location in distributed feedback control of a diffusion process. In *Proc. of the 44th IEEE Conference on Decision and Control and the European Control Conference (CDC-ECC 2005)*, pages 5662–5667, December 2005.
- [26] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.A. Fernández-Madrigal, and J. González. Multi-hierarchical semantic maps for mobile robotics. In *Proc. of the 2005 IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS 2005)*, pages 2278–2283. IEEE, August 2005.
- [27] H. Hu and D. Lee. Semantic location modeling for location navigation in mobile environment. In *Proc. of 5th Int. Conference on Mobile Data Management (MDM 2004)*, pages 52–61. IEEE, January 2004.
- [28] N. Farenc, R. Boulic, and D. Thalmann. An informed environment dedicated to the simulation of virtual humans in urban context. *Computer Graphics Forum*, 18(3):309–318, September 1999.
- [29] S. Lloyd. Least square quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982.
- [30] S. G. Nash. TN/TNBC software. http://www-fp.mcs.anl.gov/OTC/Guide/SoftwareGuide/Blurbs/tn_tnbc.html, 1984.
- [31] S. G. Nash and J. Nocedal. A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization. *SIAM Journal on Optimization*, 1(3):358–372, 1991.
- [32] Z. Di, M. Emelianenko, and S. G. Nash. Truncated newton-based multigrid algorithm for centroidal vorono diagrami calculation. *Numer. Math. Theor. Meth. Appl.*, 2011.
- [33] J.S. Wiggins. *The Five-Factor Model of Personality: Theoretical Perspectives*. The Guilford Press, New York, 1996.
- [34] F. Durupinar, N. Pelechano, J. M. Allbeck, U. Gudukbay, and N. I. Badler. How the ocean personality model affects the perception of crowds. *IEEE Computer Graphics and Applications*, 31(3):22–31, 2011.
- [35] A. Maslow. A theory of human motivation. *Psychological Review*, 50:370–396, 1943.