Self-Organizing Strategy: Adaptable Growth Model For Architecture And Urban Design

TARO NARAHARA
New Jersey Institute of Technology

INTRODUCTION

In recent years, many scientists have started to gain the advantages of self-organizing systems in nature through their computational models in areas such as telecommunication networks and robotics. These systems display advantages such as robustness, flexibility, and adaptability over many conventional systems.[1] Self-organizing computation is a computational approach that brings out the strengths of the dynamic mechanisms of self-organizing systems: structures appear at the global level of a system from interactions among its lower-level components. In order to computationally implement the mechanisms, the system's constituent units (subunits) and the rules that define their interactions (behaviors) need to be described. The system expects emergence of global-scale spatial structures from the locally defined interactions of its own components.

In past years, the author has conducted agent-based simulations to investigate collective behaviors seen in pedestrians and has studied emergent phenomena such as lane formation. "Lane formation" is a fascinating emergent phenomenon we can observe from simple agent-based pedestrian simulation (figure 1). In crowds of oppositely walking pedestrians, the gradual formation of varying lanes of pedestrians moving in the same directions are observed. This is an empirically observed collective phenomenon and has been recorded in many real-life locations such as crowded pedestrian streets crossing in the city of Tokyo.[2] The emergence of this spatiotemporal pattern is a result of non-linear interactions among pedestrians, and groups of

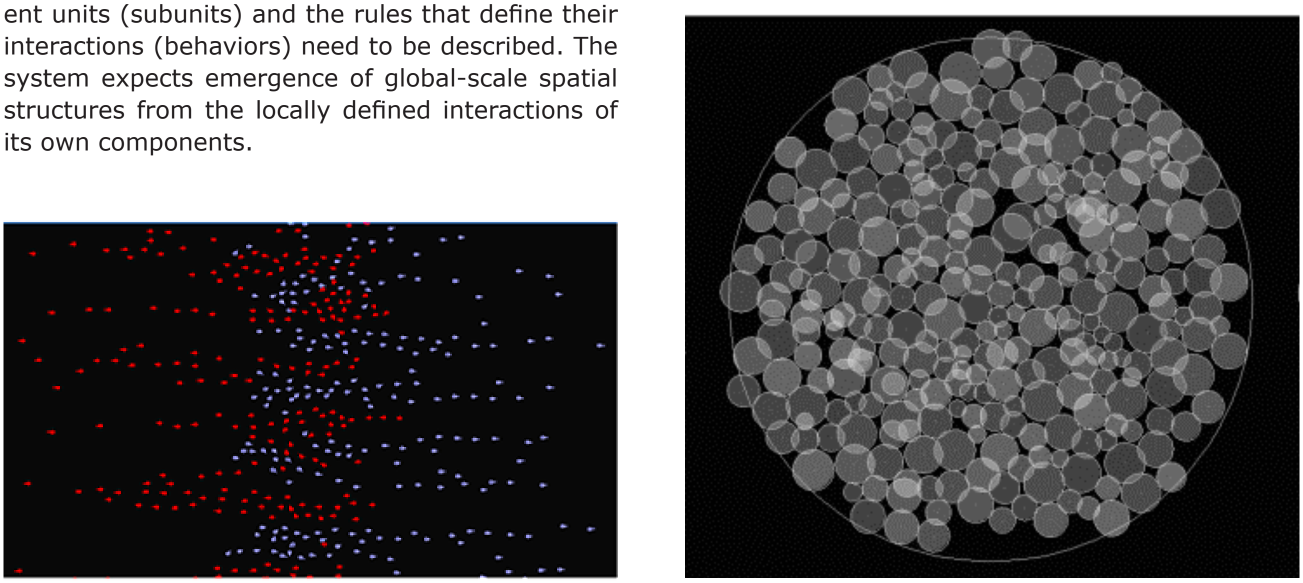

Figure 1. Pedestrian simulation by the author and emergence of lanes (left). Circle packing using a Bubble Mesh Method (right).

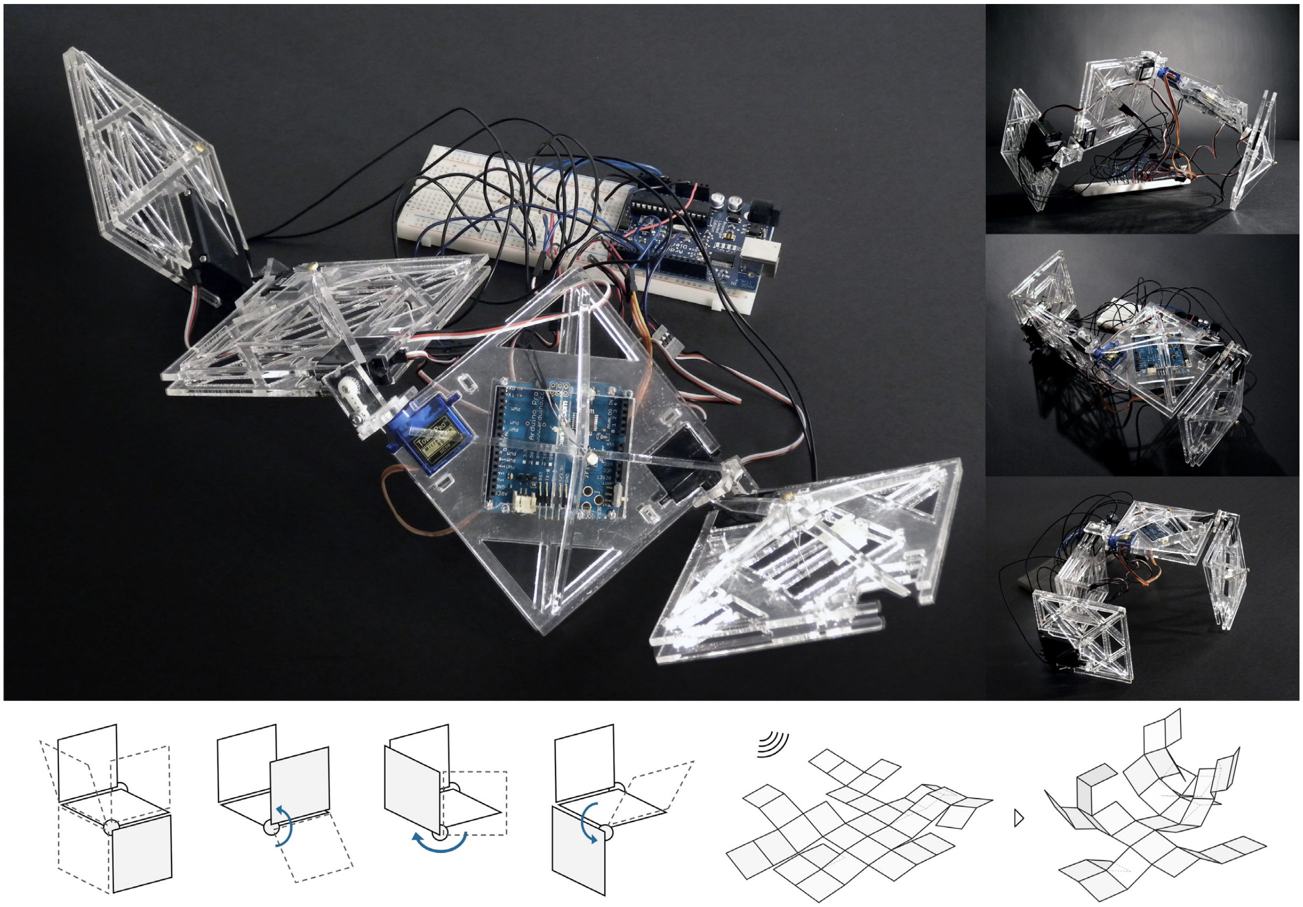

Figure 2. A robotic device with locally embedded sensors and microcontrollers.[3]

pedestrians can find efficient walking formations solely from locally embedded individual behaviors without imposing any global geometry.

Another example of self-organizing computation using local interactions by agents is close packing of circles within a circle. Circle packing is one typical

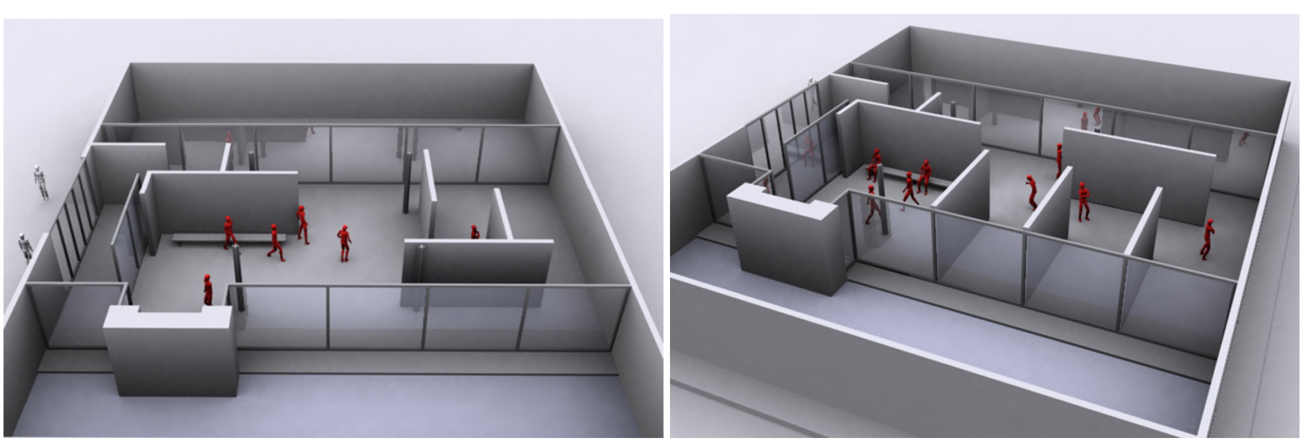

Figure 3. Simulation and visualization of human behavior by the author.[4]

case where simulation using dynamics excels the performance of any analytical means, and it can be implemented by relatively straightforward codes. Simple, locally implemented physical motions of bubbles – pushing and squeezing against each other – can eventually lead a group of bubbles to form a globally cohesive structure. The author has further explored these characteristics in the context of architecture. Figure 2 shows a robotic device with locally embedded sensors and microcontrollers. Bottom-up control strategies allow the device to optimize its orientation with respect to a light source, independent of how and where the unit is placed. Similar bottom-up strategies can be applied to a wide range of digital and physical applications.[4] This paper further explores an application of self-organizing logics in generative design systems on an urban scale.

BACKGROUND

Within the wide range of application areas in architecture, computational experiments using pedestrian agents will be studied in this paper. In the past, the author mainly focused on visualization and simulation of human behaviors inside architecture space through agent-based computation. Its goal was to add a sense of place to the geometry and to augment the representation of its spatial quality for designers and audience. Spatial qualities in architectural design cannot be fully evaluated solely by observing geometrical constructs without reference to inhabitants placed inside. (Figure 3,)[5] In this paper, beyond the simple visualization and simulation of human behaviors, computational simulation of spatio-temporal growth processes using pedestrian agents will be discussed.

There are earlier works that introduced methods to produce spatial patterns using computational agents' behaviors collectively. Helbing et al.[6] have done computational simulations using their "active walker model" and simulated trail formations observed within a human trail system. Their agents use a marking behavior that leaves modifications of the ground that makes it more comfortable to walk on. This implementation is similar to formation of trail systems by certain ant species using chemotaxis, and an algorithmic implementation of chemotaxis was also introduced by Doringo.[7] In the architectural research area, Schaur[8] investigated empirical precedents of a human trail system, and these precedents indicate that many computational simulation results are able to obtain several characteristics of pattern formations in a human trail system. This paper further explores agents' behaviors in the context of architecture and urban design and proposes a computational method that simulates growth processes of settlement patterns in time series.

METHOD

This paper further explores a generative aspect of agent-based models and proposes a computational method that simulates growth processes of settlement patterns in time series. The proposed system's fundamental components are terrain, agents, and buildings. The system assumes that the development starts from unoccupied empty terrain, and wandering settlers' behaviors are simulated by the computational agents. The buildings and streets (trails) will be gradually inserted by the agents as a part of their behaviors based on topographical information as a primary input. The entire system is written in the Visual C# language, and it has an ability to import any surfaces from various applications such as Google Earth by exporting and reading the DirectX file format.

The original environment of the system is completely vacant: unoccupied. The first settlers arrive at a site from some distance away, at some point in its history. Some settlers might be merely passing by the site, but some may settle down somewhere inside the site. As more settlers migrate through the site, trails gradually emerge. Along these trails, people find preferred locations for their shelters and begin settling down to form clusters of dwellings. Some trails start to bundle together to form arteries, and some branch out to form a complex network of passages.

Geometrical conditions of landscape are described as a grid of triangulated patches, and they can store dynamic local information about traffic intensity of agents (settlers) and local acuteness of a terrain (slope). The agents' primary behavioral characteristics include physical mobility such as hill-climbing ability, their attraction toward environmental conditions, their selection of paths based on a local traffic density, and presence or absence of global destinations. These primary factors that govern agents' heading direction vectors are named, respectively, as attraction to gentle slope, attraction to traffic intensity, and attraction to destinations.

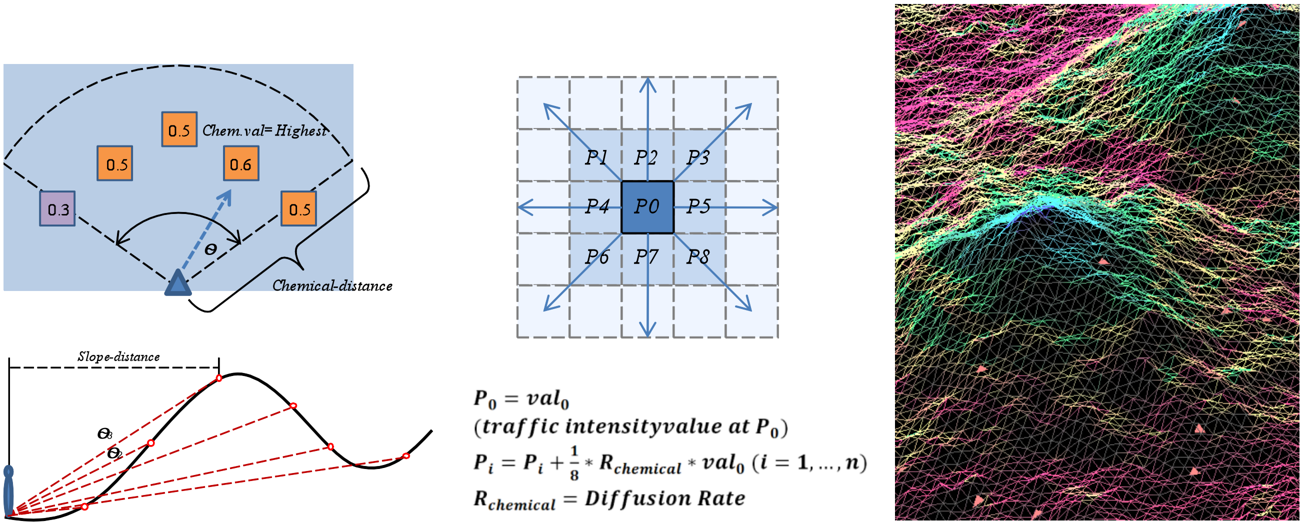

Figure 4. Agents on a wire-framed terrain with color-coded traffic frequencies (right). Slope recognition by agents (bottom-left). Agent's cone of vision & chemical value checking mechanism (top-left). Diffusion rate applied to surrounding patches of agents (middle).

The system advances every finite step of time. Every step of simulation, the reduction rate is applied to the traffic intensity value of all subdivision surface areas. The value for attraction – intensity of traffic – is directly related to the visibility of the trails, so that less-used trails will eventually fade away. Traces of trails by agents are perceptible by others in the region at adjacent patches and attract them. Trails that are frequently used by agents become attractors for agents in their neighborhoods. Consequently, the frequency of these trails' usage is amplified. The positive and negative feedbacks work as a pair to produce organic gradual transformations of street patterns such as branching, bundling, and emergence of hierarchy among them.

In many real-life scenarios, the above three attractions can coexist simultaneously and can be applied together to each agent by assigning different weights for each resulting directional vector for the three different attractors. These three weight factors are named slope-factor (s-factor), traffic-intensity-factor (t-factor), and destination-factor (d-factor).

Vagent = s * Vslope + t * Vtraffic + d * Vdest

(s + t + d =1.0; 0 ≤ s, t, d ≤ 1.0)

A normalized sum of three vectors weighted by the above-mentioned factors produces the heading vector of an agent. By manipulating the proportions of these three factors, agents' behaviors can be directed. These values are also dynamically changeable parameters based on changing environmental potentials in later experiments. In these preliminary scenarios with fixed agents' behaviors throughout the run of simulations, the results are categorized in roughly three emergent patterns, direct paths, minimal ways, and detours, and other in-between patterns.

The high value for slope–factor induces agents to find a comfortable walking path. By avoiding climbing or descending a steep hill, agents produce detours. The terrain has two steep hilltops at the south and north, and detours around these hilltops are recognized from the results. The high value for destination-factor induces agents to find the shortest path to form a direct path system. The high value for traffic-intensity-factor stimulates agents to minimize overall length of circulation. When agents have more frequent trips between destinations, shorter overall length of the system is beneficial due to its lower construction costs of roads: minimal ways. These three factors can be applied in various different proportions to find compromise solutions among three different motivations.

One of the biggest merits of the system is that purely microscopic behaviors can produce a global

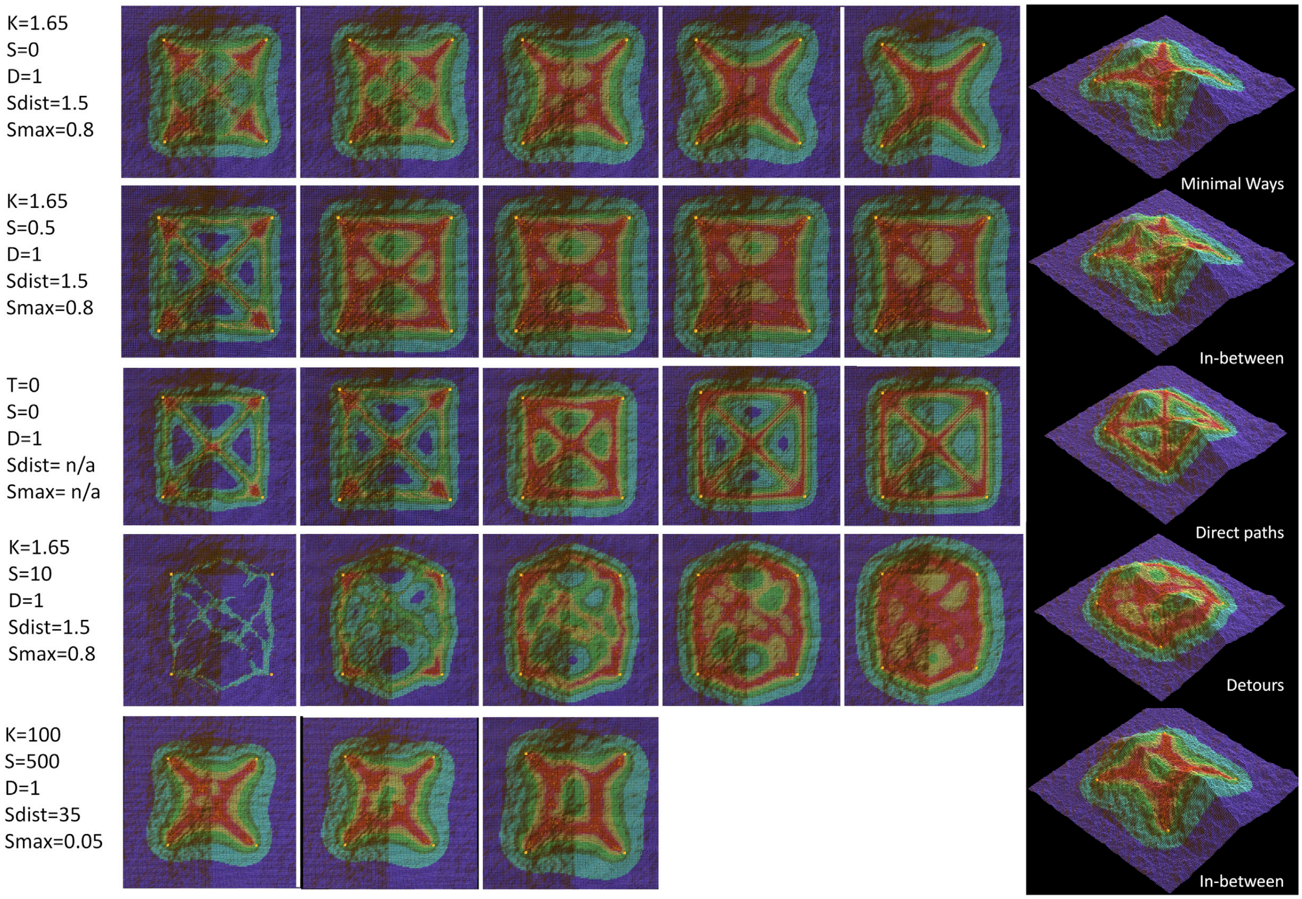

Figure 5. 4 Cities on uneven terrain: Results with various values for S, T, and D factors. D is set as a constant value 1.0 above. (Actual D value in percentage =D/(D+S+T)*100)

configuration without requiring any macroscopic information to be input into the system. When numbers of cities are small, resulting configurations are rather predictable. However, deriving a street configuration from a large number of cities at arbitrary locations on irregular terrain geometry is a challenge. Multi-agent simulation is not necessarily the fastest method of derivation, but it is a reasonably robust method for deriving street configuration as it only requires microscopic behaviors as input information.

GROWTH SIMULATION

The system can change subunits' behaviors over time based on stimuli from the changing environment. In the following section, shifts in agents' behaviors stimulated by environmental changes are considered. In order to evaluate the system, information from an existing site, San Miniato in the region of Tuscany in Italy, was used due to its unique correlations between its landform and urban settlements. Comparisons of results from the system and the actual existing site are presented.

Once an entire environment's activity level reaches a certain maturity, agents start to rely on information that already exists in environments. Agents start to follow higher traffic frequency areas instead of trails that no one has been taking (i.e., t-value increases). The above behaviors were implemented by setting a threshold value to shift agents' behaviors. As the number of terrain patches that possess traffic intensity value of over 0.8 exceeds 15% of the total number of patches, agents start to check the traffic frequency around them to make decisions about their heading directions.

After the emergence of street networks, some of the intersections of several arteries become popu-

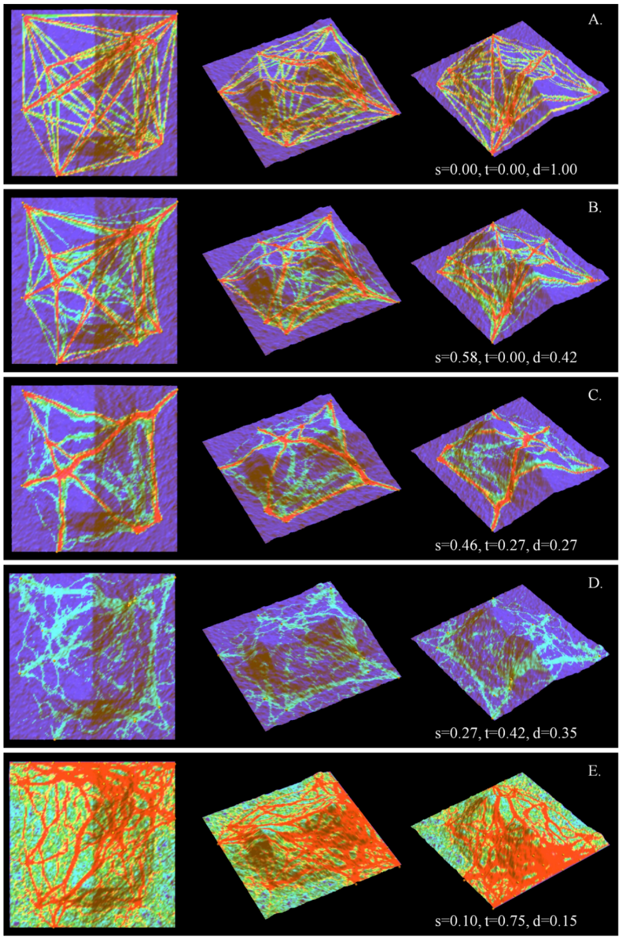

Figure 6. Results of emerging patterns on a terrain with 8 predefined stationary destination points and two hills with various values for s, t, and d factors.

lation concentration areas, and these areas have the potential to grow into cities. As the number of terrain patches that possess traffic intensity value of over 0.8 exceeds 18% of the total number of patches, the system starts to check for and find traffic-intensive areas. The algorithm finds peak areas of traffic intensity value above a certain threshold value, and finds places where these peaks are forming clusters. Traffic-intensive patches that are within a certain distance from each other are read as one island. If these islands are larger than a certain minimum size, they are considered as city areas. After sufficient careful trials, numbers that produce results that represent a natural scale of development relative to the scale of terrain are empirically adopted. Once cities are registered by the system, all agents will travel around these cities.

Emergence of buildings is dependent on environmental potentials that are mainly frequencies of traffic and topographical conditions of a terrain. The frequencies of traffic at each patch of the terrain are considered as an indication of population density. When this value is higher than a certain threshold value, the site becomes a potential location for buildings under a certain probability. The sizes and heights of the buildings are defined by a negative exponential function of traffic intensity value to reflect upper-bound heights found in real-world settlements.

Slopes of the site are another criterion for building sites. If the maximum slope of the site is above a certain degree, it is quite likely that no settler is willing to build any structures at such a steep slope. This maximum value can be dependent on regional tectonic cultures, available materials and technologies, and climatic conditions. I set this to 30 degrees based on conventional standards as a maximum slope angle for buildable areas. Buildings have an ability to align themselves to agents' heading directions and shift their locations to avoid direct collisions. Buildings also calculate the average rotation angle of others within a certain distance away and try to align themselves to the average angle. This is a self-organizing behavior often seen among a flock of birds. Synchronization among buildings' rotation angles is eventually expected to produce natural arrays of buildings.

Once the overall environmental potential grows above the aforementioned minima, cities will have emerged. At this stage, settlers are no longer randomly wandering migrants seeking temporary shelter. Instead, they act as heterogeneous self-driven agents based on clear objectives of their own and travel through these newly emerging cities. Spatiotemporal developments of the site are organized by the settlers' behaviors; however, the environmental changes induced by the settlers also simultaneously influence their behaviors. Figure 6-left shows a gradual growth process of street networks and settlement patterns that have captured characteristics that exist in the current actual site condition of San Miniato, and Figure 6-right shows five schemes with different parameter settings that display extreme urbanistic future scenarios for San Miniato in a speculative domain.

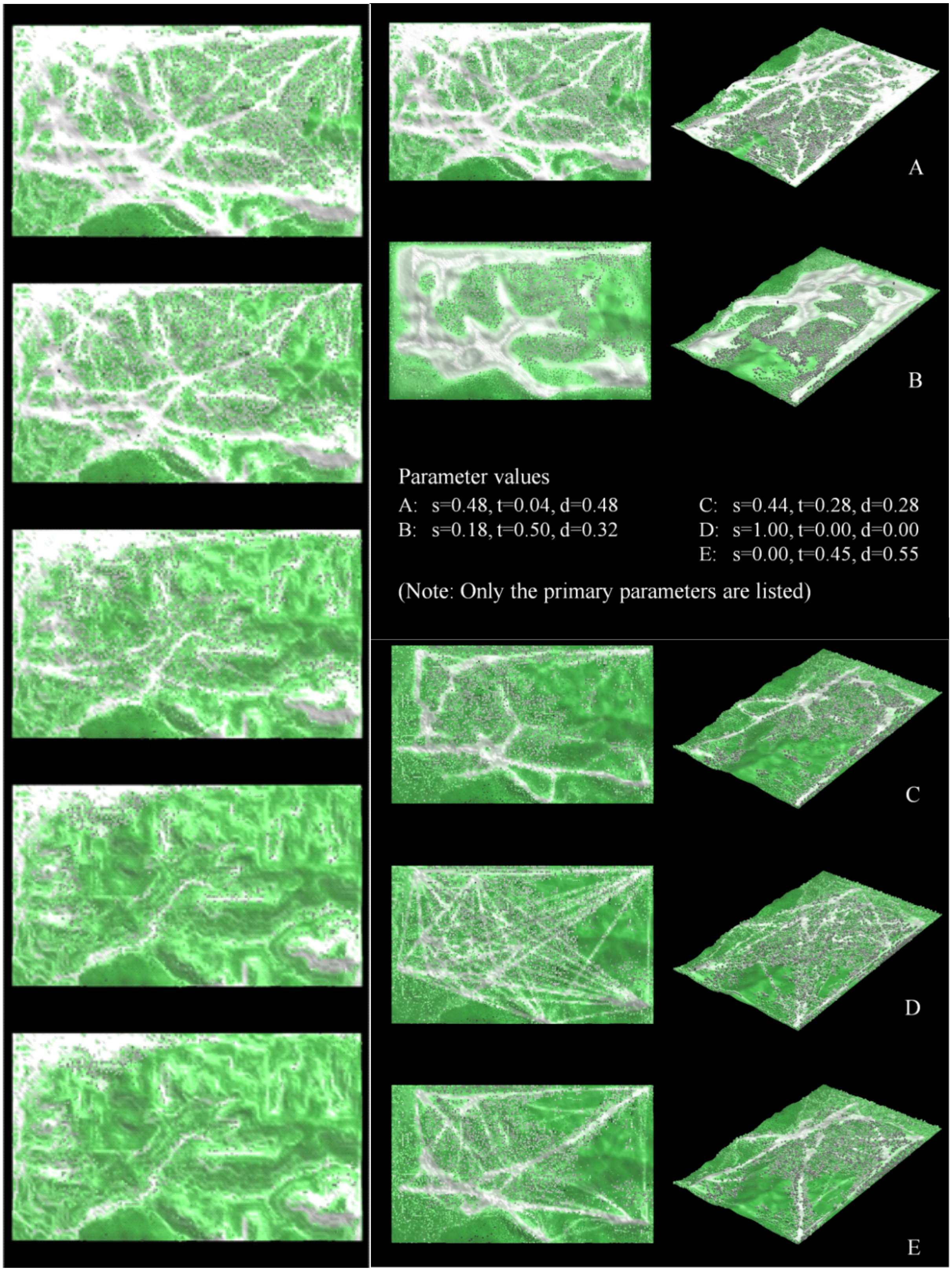

Figure 7. City Growth phase with scheme A: Gradual growth of paths and buildings (Left). City Growth Phase with 5 different parameter settings (Right).

CONCLUSION

The system outlined here uses interactions and feedback among its own components, agents and environment and produces new instances of spatial layout of paths and buildings from primary inputs of a given landform and environmental conditions. Agents' behaviors are updated accordingly as new paths and buildings are generated. This co-evolutionary process between agents and environments is known to exist in many self-organizing systems.

In conclusion, there are several advantages of self-organizing computation in development of generative design systems in urban design. Firstly, the most obvious advantage is that self-organizing computation can represent and generate growth processes over time. This means that the approach can design a system in transition, and it is a necessary feature for simulating decentralized dynamics of settlements.

Secondly, one of the byproducts of a growth model is the ability to produce new objectives based on the model. Discovering unknown design objectives (in a sense, the emergence of new architectural programs) is one unique characteristic that is potentially applicable to planning and strategic development of architectural projects in earlier stages.

In this example, environmental growth of the system and agents' behavioral changes lead the entire system to gradually adapt itself to emerging states of the system. "Developing shortest path patterns for agents to travel around cities," "creating paths that have least elevation changes for agents' trips," or "building allocations that can maintain functional traffic patterns" are several interpreted objectives from the results of the system with various different parameter settings. These objectives are ex post interpretations of the results and are not provided directly as initial requirements or goals for the system. During the course of the system's run, these changes toward satisfaction of certain objectives emerge as a result of the system's ability to adjust its response to stimuli according to the state of the environment.

Finally, one of the unique characteristics of self-organizing computation is its non-reliance on any external knowledge. As with many conventional computational methods in architecture, such as L-system and shape grammar, imposing existing design patterns or transformation sequences is beneficial when one wants to efficiently derive what appear to be the subjects of our recognitions. However, reliance on a pre-existing template might preclude the possibility of discovering what original inputs naturally turn into.

In this conceptual experiment, any imposition of knowledge from outside of the system has been thoroughly excluded from the process. Instead, the knowledge is acquired from the process of development over time, and implementation of the knowledge is conditionally applied in continuous sequences. What is separating the results from direct paths, detours, or minimal ways is a simple set of parameters. Instead of providing predefined design templates such as grid, radial, or branching patterns, sets of parameter values that generate agents' behaviors are driving the resulting configurations. In this way, the inherent characteristics of the resulting configurations are traceable back to several parameter values that govern the behavior of the system, and certain sets of parameters that lead to characteristics similar to existing urban phenomena can be studied. The goal of the experiment is to derive forms from behaviors instead of supplying a formal knowledge of design patterns at the outset.

ACKNOWLEDGEMENTS

I would like to sincerely thank my former academic advisers, Professor Martin Bechthod and Professor Kostas Terzidis at Harvard University, and Professor Takehiko Nagakura at the Massachusetts Institute of Technology, for their insightful guidance and constant support. I would also like to thank my current employers, Dean Urs Gauchat and Professor Glenn Goldman at New Jersey Institute of Technology for their generous academic support. Finally, I would like to thank the organizers of the ACSA Conference for this great opportunity.

ENDNOTES

1 Camazine, Deneubourg, Franks, Sneyd, Theraulaz, Bonabeau, (2002). Self-organization in Biological Systems, Princeton University Press. Princeton, New Jersey.
2 Katoh, I. et al. (1980) Characteristics of lane formation, Nihon Kenchiku-gakkai Ronbun houkoku-shu, No289, p121
3 Narahara T. (2010). Self-organizing Computation: A Framework for Generative Approaches to Archi-

tectural Design, Doctor of Design dissertation, Harvard University Graduate School of Design, Cambridge, MA.
4 Narahara T. (2010). Design for Constant Change: Adaptable Growth Model for Architecture, International Journal of Architectural Computing (IJAC), IJAC 8-1.
5 Narahara T. (2007). The Space Re-Actor: Walking a Synthetic Man through Architectural Space, MS thesis, Massachusetts Institute of Technology, Cambridge, MA.
6 Helbing, D., Farkas, I. J., and Bolay, K. (2001). Environment and Planning B, 2001, Self-organizing pedestrian movement, Environment and Planning B: Planning and Design 2001, volume 28, pages 361- 383.
7 Dorigo, M. (1992). Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milan, Italy.
8 Schaur, Eda, (1991). Non-planned settlements: Characteristics features – path system, surface subdivision, Editor Frei Otto, volume 39 of Institute for lightweight structures (IL), University of Stuttgart, Germany.
9 Fanelli, G. (1990). Toscana, Francesco Trivisonno Firenze : Cantini.