EUROGRAPHICS 2015 / O. Sorkine-Hornung and M. Wimmer
(Guest Editors)

Volume 34 (2015), Number 2

Biologically-Inspired Visual Simulation of Insect Swarms

Weizi Li [1] David Wolinski [2] Julien Pettré [2] Ming C. Lin [1]

[1]University of North Carolina at Chapel Hill, USA
[2]Inria Rennes, France
http://gamma.cs.unc.edu/VisInsectSwarms/

Figure 1: *Simulation of butterflies moving on a prairie.*

Abstract

Representing the majority of living animals, insects are the most ubiquitous biological organisms on Earth. Being able to simulate insect swarms could enhance visual realism of various graphical applications. However, the very complex nature of insect behaviors makes its simulation a challenging computational problem. To address this, we present a general biologically-inspired framework for visual simulation of insect swarms. Our approach is inspired by the observation that insects exhibit emergent behaviors at various scales in nature. At the low level, our framework automatically selects and configures the most suitable steering algorithm for the local collision avoidance task. At the intermediate level, it processes insect trajectories into piecewise-linear segments and constructs probability distribution functions for sampling waypoints. These waypoints are then evaluated by the Metropolis-Hastings algorithm to preserve global structures of insect swarms at the high level. With this biologically inspired, data-driven approach, we are able to simulate insect behaviors at different scales and we evaluate our simulation using both qualitative and quantitative metrics. Furthermore, as insect data could be difficult to acquire, our framework can be adopted as a computer-assisted animation tool to interpret sketch-like input as user control and generate simulations of complex insect swarming phenomena.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

1. Introduction

Representing the majority of living animals, insects are the most ubiquitous biological organisms on Earth. Fascinated by their collective behaviors, researchers from various disciplines have studied intrinsic operating mechanisms of insect swarming phenomena with various aims, such as health and food issues, but also more surprisingly novel methods to facilitate design, architecture, construction, and optimization of many complex structures. In graphics research, with increasingly more realistic animation of various life forms, simulation of insect swarms can enhance visual realism of many graphical applications from virtual environments, computer games, to cinematography.

The fascination with insect swarms lies in the very complex and diverse nature of insect behaviors *as individuals* and *in aggregation*. This variety, arising from the heterogeneity in sensing and cognitive abilities, survival strategies, interpretations of environmental factors, etc [CDF*03], makes simulation of insect behaviors one of the most challenging computational problems. A complete model to capture so different and complex phenomena seems infeasible and can introduce an unmanageable amount of parameter tuning and tweaking. To overcome this difficulty, we propose a general biologically-inspired framework to analyze, evaluate and simulate insect swarms. We also evaluate simulation results both qualitatively and quantitatively.

Swarming of insects often exhibits complex global patterns as a whole, yet frequently combines local randomness and chaotic individual movements. Motion of insect swarms can be generally decomposed into different spatial scales [Oku86]. At the microscopic scale, insects have local interactions among them (e.g. collision avoidance). At some intermediate mesoscopic scale, insects move in some preferred direction and form trajectories of various shapes. At the macroscopic level, more coherently, global structures emerge from collections of individual movements. What drives the overall direction of an insect swarm has been conjectured due to various reasons, including food seeking and hunting, predator avoidance and evasion, courtship and mating, and seasonal migration, etc.

Given the complex causalities leading to insect swarming, we propose a multi-scale approach that is biologically inspired [Sum10] and experimentally validated. Our objective is to recreate realistic visual simulation of insect swarms by learning from real insect motion data. We analyze the insect motion data at micro-, meso- and macroscopic scales to configure and estimate parameters of various components residing in our simulation framework. More specifically the captured data enable us to (1) automatically select and configure the most suitable local steering algorithm, (2) generate statistically accurate waypoints at mesoscale using segmented piecewise-linear trajectories, and (3) parametrize a global guiding field that captures swarming spatial patterns.

Since insects motion data are not always available and sometimes animators prefer to have direct control over the resulting simulation, our framework can further serve as an assistive animation tool that empowers artists and amateurs. Our prototype system can interpret sketch-like input and generate similar patterns; or the user can provide "guiding simulation results" with specifications on exploited motion models and related parameters, our simulation can then *re-target* the motion onto different swarming behaviors.

In summary, the main contributions of this work are the following:

- A general biologically-inspired framework is developed to analyze, evaluate and simulate insect swarms (Section 3).
- Comparative evaluations of various local steering algorithms are performed to select and configure the best model for insects local collision avoidance (Section 4.1).
- A statistical learning and evaluating approach is developed for modeling individual behaviors within the swarm and preserving spatial structures of the swarm (Section 4.2 and 4.3).
- The resulting framework can serve as an assistive animation tool for users to either turn sketched patterns into animations or utilize existing simulations for re-targeting purposes (Section 5).

2. Related Work

Insect swarms can be interpreted as multi-agent systems. Multi-agent simulation has been a flourishing research topic for nearly three decades. Many aspects including local collision avoidance, global path planning, behavior modeling, group motion and user interaction have had striking results. A large portion of these problems have been addressed under the focus of crowd simulation. In this section, we will first discuss models that proposed for generating agents' local and global behaviors, then proceed to models developed from biology, physics and mathematics perspectives in studying insect swarms.

At microscopic level, a variety of approaches have been developed to address the local collision avoidance problem. These include the seminal rule-based Boids model [Rey87], Social Force model [HM95], Cellular Automata model [KS02], velocity obstacle based models [GCK*09, vdBGLM11], synthetic vision model [OPOD10] and other hybrid approaches [PAB07, SKH*11]. Generally speaking, local collision avoidance is performed to steer agents away from other agents so that no or less collisions would occur and usually reflected by changes of the preferred velocity to the actual velocity. However, while each agent is treated as an independent entity and obeys local rules, the formed global motion patterns from multiple agents may not be desired.

When global behavior becomes the focus, several continuum models have been developed to navigate agents, examples including [Hug03, TCP06, NGCL09]. These approaches usually take positions and velocities of the agents and construct density and velocity fields as the guiding graph in order to simulate a large crowd. Other work have focused on providing users the flexibility to alter crowd flows. For example, Kown et al. [KLLT08] and Takahashi et al.

[TYK*09] enabled group editing capability in order to generate believable formations. Patil et al. [PvdBC*11] built a navigation field which can interpret sketch-like data as input. The most recent work done by Wang et al. [WJDZ14], combining noisy local behavior and smooth global trajectory, introduced field-based approach to simulate insect swarms. While the visual believability has been improved, in order to generate emergent global structures of the swarm, empirical design is required.

From biology, physics and mathematics perspectives, there exist two major approaches in studying and simulating insect swarms, namely individual-based models and continuum models. Individual-based models include Self-Propelled Particles (SPP) model [VCBJ*95], variants of the SPP model [CkJ*02, DCBC06], force-based models [FGLO99] and Brownian particles [SESG08]. These models, taking the Lagrangian system viewpoint, introduce local rules to each particle and focus on its response to neighboring particles or external stimuli or both. As of certain models such as the SPP model, by maneuvering local rules alone, global orders appear and sometimes a close approximation to the actual insect dynamics is observed [BSC*06]. Continuum models, instead, adopt the Eulerian representation, depict swarms in certain features such as density and velocity fields and use these features to guide member interactions, examples including [MEK99, TBL06]. Another modeling trend is based on the assumption that insect movements contain certain degrees of randomness. Thus a general macroscopic regulation cannot describe overall dynamics of an system but rather statistical values of it. Many work such as [DCBC06, YEE*09, BKBS13] built on top of this assumption and incorporated various potential functions and noise types into the simulation.

While many aspects of the animal aggregation have been studied and various models have been developed, these work mainly focus on either functional or mechanical aspects of insect swarms, to name a few, mating strategies [BMD*13], phase transitions and kinematic fluctuations [VCBJ*95, ACC*13], and population clumping phenomenon [TBL06]. The effectiveness of applying these approaches on graphical applications is unclear and comparative evaluations of these approaches seem scarce. In addition, noted by Butail et al. [BMD*13], few models are validated against reference data giving that animals motion tracking task is non-trivial. In comparison to previous work, our system takes data-driven approach, independent of insect species, automatically conducting model evaluation and configuration, and generates realistic insect swarming behaviors on multiple scales.

3. Approach Overview

In this section, we give an overview of each part of our framework. As part of the off-line process, our simulation pipeline processes all the insect motion data into a collection of linear trajectory segments and a density field. We then use these information to (1) perform automatic selection of local steering algorithms, (2) construct probability distribution functions for generating intermediate waypoints for insects traveling, and (3) capture the spatial pattern of collective swarming behaviors at the global scale. Lastly, we describe how we combine these components into the final simulation system. The process is illustrated in Figure 2.

3.1. Data Processing

Insects' trajectories have been observed to be zig-zags [Oku86]. Inspired by this observation, we divide the trajectories into successive linear segments using the piecewise linear regression [NWK90] with a goodness of fit $R^2 = 0.9$. These segments will serve to choose the best collision avoidance model and set up the mesoscopic goal-selecting algorithm.

Stating this formally, the data [KO13, PKO14] $\mathbf{z}$ can be defined as containing the agents' position information at any timestep k. Thus, assuming m timesteps are recorded in the data:

$$\mathbf{z} = \bigcup_{k=1}^{m} \mathbf{z}_k. \tag{1}$$

For a given trajectory segment s of the ith agent, the data $\mathbf{z}_{s_i}$ is:

$$\mathbf{z}_{s_i} = \bigcup_{k \in s_i} \mathbf{z}_k, \tag{2}$$

where $s_i = [s_i^1, \ldots, s_i^n]$ groups all timesteps in the linear section, starting with $s_i^1 \geq 1$ and ending with $s_i^n \leq m$.

At this phase, we exploit data frames which contain positions and velocities of insects [PKO14]. Piecewise-linear segments and turning angles were automatically extracted to establish corresponding segment length and turning angles distributions. For the experiments in this paper, we used approximately one million frames for more than 500 insects recorded at 100 FPS. Given about 100,000 extracted segments, the average segment length is 14.57 mm (median = 7.22, standard deviation = 19.01) and the average turning angles are around 2.34 degrees (median = 2.73, standard deviation = 40.02). We also discretized the simulation space into a number of cells and calculated the visiting frequency of all insects within each cell to construct a static density distribution.

While this work is inspired by zig-zag traveling fashion of common insects, as a merit of the piecewise linear regression, our approach does not require "zig-zag" trajectories. The data processing steps and simulation methods are general, independent of insect species, and applicable to smooth insect trajectories as well.

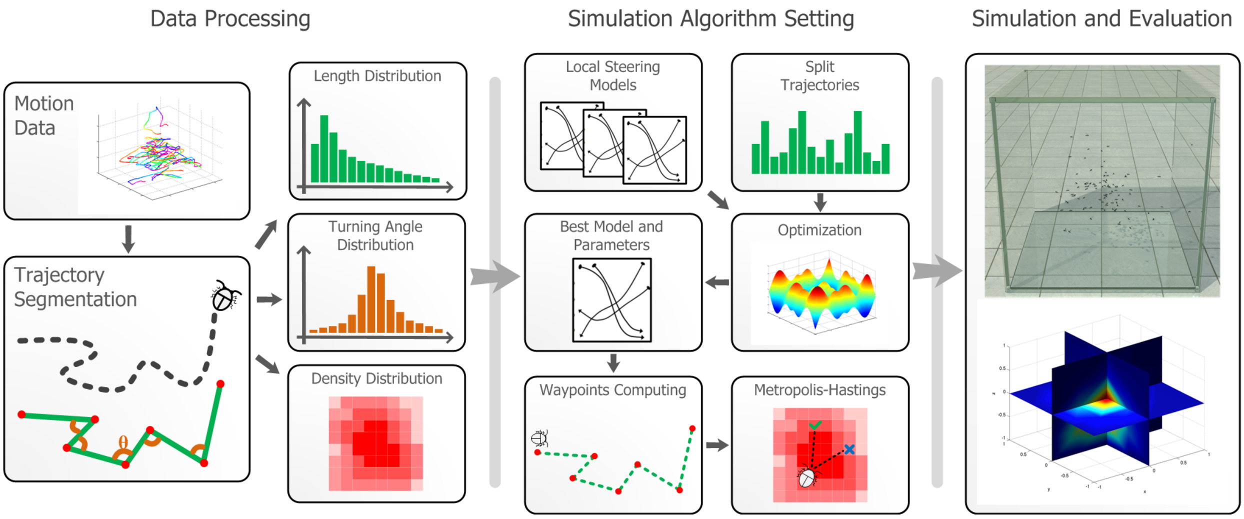

Figure 2: *The schematic view of our simulation pipeline.*

3.2. Simulation Algorithm Setting

At the lowest level of our simulation pipeline, lies the collision avoidance algorithm which ensures no insects would collide into each other. A collision avoidance algorithm can be defined as a function f which, given the current timestep k, the current positions of all agents $\mathbf{x}_k$, the agents' current velocities $\mathbf{v}_k$ as well as their goals $\mathbf{g}$, computes the agents' positions and velocities at the next timestep:

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{v}_{k+1} \end{bmatrix} = f(\mathbf{x}_k, \mathbf{v}_k, \mathbf{g}). \tag{3}$$

Our framework using optimization techniques automatically configures candidate local steering algorithms. This configuration aims at finding the best performance of each steering algorithm during approximation to the actual trajectories from the data [BKHF14, WGO*14]. After getting configured optimal parameters of each steering algorithm, we can perform comparative evaluations and determine which model is the most suitable one for simulating insect behaviors at the microscopic level.

Note that, to actually generate trajectories for each insect agent, the local collision avoidance algorithm requires a method for computing waypoints at the meso-scale. To achieve that, the segment length and angle distributions constructed at the data processing phase are sampled and used for calculating the next waypoint once an agent has reached its current one. This statistical approach aims to compensate the noisy and chaotic aspects of insect local behaviors.

However, by using only the wayfinding algorithm with local collision avoidance, there is no control over the swarming spatial structure. In order to capture emergent patterns arise from individual insect trajectories, we adopt the Metropolis-Hastings algorithm to implicitly coordinate the motion and swarming patterns. This Markov Chain Monte Carlo method guarantees that, with enough samples the overall spatial structure would resemble the reference data with respect to the density distribution. The reason we take this stochastic approach instead of explicit rule-based specifications, e.g. speed and direction alignment, separation maintenance, etc, is because rule-based approaches are more context-dependent while our objective is to develop a general framework that can work with and learn from any types of data for a diverse variety of insects.

4. Optimization and Runtime Algorithms

In this section, we describe how we use the reference data to choose the best collision avoidance algorithm for insects simulation and develop our mesoscopic and macroscopic goal-selecting algorithms.

4.1. Microscopic Reactive Behavior

The collision avoidance algorithm generally regulates agents' movements at microscopic level. Such algorithms usually have several tunable parameters (e.g. agents' size, agents' preferred velocity...) which can influence the simulation results. With all agents' parameters $\mathbf{p}$, a collision avoidance algorithm f (Equation 3) can be extended to the parameterized version as:

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{v}_{k+1} \end{bmatrix} = f(\mathbf{x}_k, \mathbf{v}_k, \mathbf{g}, \mathbf{p}). \tag{4}$$

Considering a parameterized collision avoidance algorithm, a simulation on a linear section $\mathbf{z}_s$ of the data at the meso-

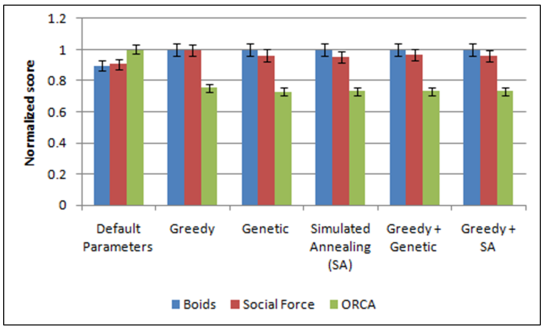

Figure 3: *Evaluation results of three steering algorithms under the Progressive Difference metric with various optimization techniques.*

scopic level can be defined as follows:

$$f(\mathbf{z}_s, \mathbf{p}) = \bigcup_{k \in s} f(\mathbf{x}_k, \mathbf{v}_k, \mathbf{g}, \mathbf{p}), \tag{5}$$

initialized with $\mathbf{x}_{s_{start}} = \mathbf{z}_{s_{start}}$, $\mathbf{v}_{s_{start}} = speed(\mathbf{z}_{s_{start}})$ and $\mathbf{g} = \mathbf{z}_{s_{end}}$.

Now, given a distance metric $dist()$ between the data $\mathbf{z}_s$ and the simulation $f(\mathbf{z}_s, \mathbf{p})$, we seek the parameter set $\mathbf{p}^{\mathbf{opt}}$ that minimizes this distance:

$$\mathbf{p}^{opt,f} = \operatorname*{argmin}_{\mathbf{p}} dist(f(\mathbf{z}_s, \mathbf{p}), \mathbf{z}_s). \tag{6}$$

Once this parameter set is found, we can rank collision avoidance algorithms. For two such algorithms f_1 and f_2, we can say that f_1 is better than f_2 if and only if:

$$dist(f_1(\mathbf{z}_s, \mathbf{p}^{\mathbf{opt,f_1}}), \mathbf{z}_s) < dist(f_2(\mathbf{z}_s, \mathbf{p}^{\mathbf{opt,f_2}}), \mathbf{z}_s). \tag{7}$$

To perform what is described in Equations 6, 7, we rely on the framework described in [WGO*14]. The framework has adopted several widely-used combinatorial algorithms including *Greedy*, *Genetic*, *Simulated Annealing* and hybrid versions of these algorithms to handle high-dimensional tasks such as the multi-agent simulation.

To evaluate insect movements and derive optimal parameters of a model that can provide the best approximation of a simulation to the reference data, we use the microscopic metric *Progressive Difference*. This metric measures the difference between the simulated and the reference trajectory between the starting and ending points of a segment while resetting the simulation to configurations of the reference data at each time step. The reason for us to pick this metric is due to the data contain many frames representing long and split trajectories (not losing track of insects during the motion capture is very challenging). Thus, to learn the collision avoidance behavior of the insects, it makes sense to focus on decisions being made at any given moment with limited influence from the past states of the swarm.

The specific steering algorithms we are evaluating are Boids model [Rey87], Social Force model [HM95] and ORCA model [vdBGLM11]. In particular, for Boids model, we only examine its collision-avoidance mechanism not other flocking rules. Since the distribution of evaluation scores appears to be non-Gaussian, we choose non-parametric within-subject tests to analyze differences between these models. The evaluation results shown in Figure 3 depict the median of scores across different steering algorithms before and after various optimization techniques have applied. As all metrics are measuring the distance between simulated and reference trajectories, smaller scores indicate better models.

By using default parameters, a Friedman test revealed a significant effect of different steering algorithms on their scores ($p < 0.01$). A post-hoc test of pairwise comparisons using Wilcoxon signed rank tests with Bonferroni correction showed significant differences between all paired groups ($p < 0.01$). As results, Boids is better than Social Force and latter is better then ORCA. We interpret this result as Boids' collision-avoidance is more reactive (direct dependence on agents' positions) than other two algorithms.

However, after applied various optimization techniques, across all evaluations, the Friedman test revealed a significance effect on different models and the post-hoc tests showed no significance between Boids and Social Force ($p > 0.01$), but significant difference between Boids and ORCA, and Social Force and ORCA ($p < 0.01$). These results indicate, after optimization, the Boids and Social Force model perform relatively the same, while ORCA outperforms both by taking into account both positions and velocities as part of its anticipatory capabilities.

To further measure the relationship between number of timesteps that a segment contains and its corresponding evaluation score. We performed correlation and causality tests. To be more specific, a Spearman's rank test shows that these two variables are very weakly correlated (Spearman's $\rho \approx 0.27$, $p < 0.01$) and constructed linear models show no significant effect of one variable on predicting the other (Intercept $p > 0.01$ and Slope $p > 0.01$). These analyses indicate that the two variables are in general independent of each other. Thus we can use a specific steering model consistently rather than a mixture of steering algorithms switched at certain timesteps. The latter option may provide marginal improvement but at the cost of efficiency. Based on these results, we use ORCA for the local collision avoidance task with its statistically optimal parameters: *preferred speed* $(m.s^{-1}) = 1.48 \in [1,2]$, *radius (m)* $= 0.45 \in [0.2, 0.8]$, *neighbor distance (m)* $= 14.84 \in [10, 20]$, *time horizon (s)* $= 5.27 \in [3, 7]$.

4.2. Mesoscopic Waypoints Computation

Once we have chosen the best local steering algorithm, we need to compute waypoints at mesoscopic level and the seg-

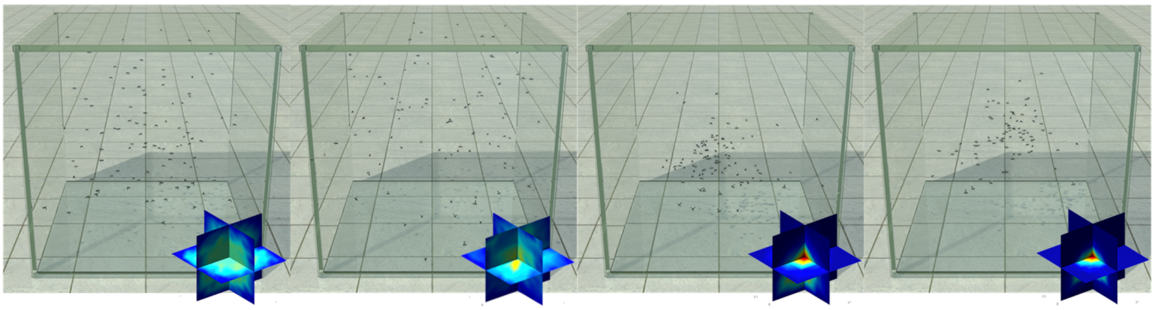

Figure 4: *Comparison of simulations using different sampling techniques to the reference data. Accumulated density distributions during the simulations are also shown.*

ment length and turning angle distributions are utilized. To be specific, let *sampleLength* be a function which returns a sample from the segment length distribution. Similarly, let *sampleTurn* be a function which returns samples from the turning angle distributions. For each agent i, at each timestep k, l_i is the length of sampled segment, $\mathbf{o}_i$ is the calculated direction by using sampled turning angles, $\mathbf{x}_{i,k}$ is the agent's current position, and $\mathbf{s}_i$ is the agent's starting position. The next waypoint $\mathbf{g}_{i,k}^{meso}$ of agent i at timestep k is calculated as in Algorithm 1.

Algorithm 1: Waypoints Computation.

if $\| \mathbf{x}_{i,k} - \mathbf{g}_{i,k}^{meso} \| \leq \varepsilon$ **then**
- $l_i \leftarrow sampleLength()$
- $\mathbf{d}_{i,k} \leftarrow \mathbf{x}_{i,k} - \mathbf{s}_i$
- $\mathbf{o}_i \leftarrow direction(\mathbf{d}_{i,k}, sampleTurn())$
- $\mathbf{s}_i \leftarrow \mathbf{x}_{i,k}$
- return $\mathbf{g}_{i,k}^{meso} \leftarrow \mathbf{s}_i + l_i.\mathbf{o}_i$

else
- return $\mathbf{g}_{i,k}^{meso}$ unchanged

end

4.3. Macroscopic Swarming

The sole computation of waypoints at mesoscopic level would result in the agents' dissemination over the whole simulation space. To improve this, we need a method at macroscopic level to preserve the insects' spatial pattern and density distribution which are essential aspects of insect collective behaviors [CDF*03]. We have adopted the Metropolis-Hastings criterion to achieve this purpose. This criterion makes no assumption of underlying population density distribution which is ideal for working with any types of motion data, especially given that different insect species have different aggregating strategies and result at complex density distributions within the overall spatial structure.

The intermediate waypoints are validated as follows. To begin with, we construct the density field found in the data by discretizing the simulation space and computing the visiting frequency of each cell by all insects. Each cell c then has a frequency w_c. Later, during simulation, when a new mesoscopic waypoint is computed and proposed, we check this waypoint with the Metropolis-Hastings criterion. That is, if $\mathbf{x}_{i,k}$ is agent i's current position, $\mathbf{g}_{i,k}^{meso}$ is its proposed new waypoint, c_x is the cell corresponding to $\mathbf{x}_{i,k}$ and c_g is the cell corresponding to $\mathbf{g}_{i,k}^{meso}$, we compute $\alpha = \frac{w_{c_g}}{w_{c_x}}$. and accept the proposed waypoint with probability $min(\alpha, 1)$. If the proposed waypoint gets rejected, another waypoint would be sampled and generated [CG95].

This criterion guarantees agents to aggregate at regions with higher densities and lead to the overall swarming phenomenon. We can also easily handle obstacles by assigning a zero frequency to the cells that are occupied by said obstacles. Not surprisingly, the discretizing precision of the simulation space affects simulation results. The analysis of this effect is elaborated in Section 5.2.

5. Results

In this section, we show our simulation results illustrated in several examples and provide both qualitative and quantitative evaluations.

5.1. Simulation

Our first example (Figure 4) shows the simulation of 100 insects moving in a confined space. This setting aims at mimicking the capturing environment of the collected data [PKO14]. In this example, we compare simulations of random sampling approach, meso-scale sampling approach, our combined meso- and macro-scales sampling approach, to the reference data. Using random sampling method, agents' trajectories tend to be very smooth; after applying mesoscale sampling method, agents start to exhibit swerving

Figure 5: *(Left) Simulation of 200 midges swarming around a lantern. (Right) Simulation of 50 butterflies moving on a prairie.*

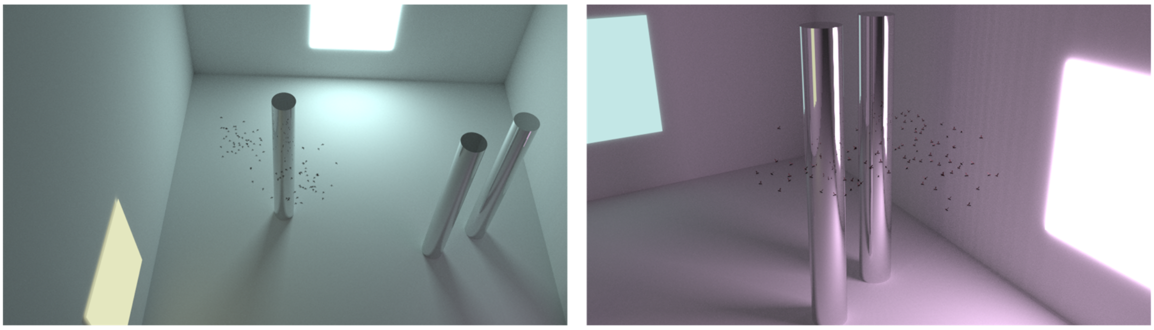

Figure 6: *A swarm of insects is traveling to successive light sources while avoiding obstacles.*

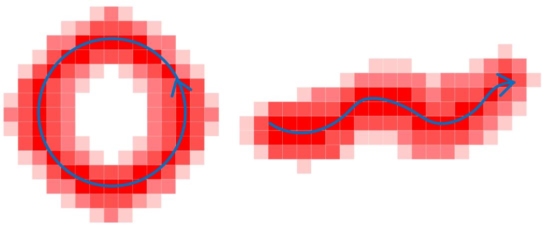

Figure 7: *Example sketches (3D) with density fields derived from the Gaussian distribution.*

and zig-zaging behaviors. However, since there exists no macroscopic-level guidance for governing the spatial pattern, agents would eventually spread out the simulation space. By adding the macroscopic level method, our technique finally shows close approximation to the reference data in preserving the swarming spatial pattern. The accumulative density information during entire simulations is also shown in Figure 4.

As we mentioned in previous sections, our framework can also be adopted as an assistive animation tool to interpret sketch-like input and generate corresponding simulation. By combining features learned from the captured data (e.g., segment length and turning angles distributions) and user specifications including sketched trajectory and other settings (e.g., density distribution, steering models and parameters). Our framework is capable of creating insects simulation of the same or variable sizes – even different species of insects.

To better demonstrate, we embedded the simulation in virtual environments†. The first example (Figure 5 Left) is showing by interpreting a circle sketch at the left side of Figure 7, we get sampled trajectories – within density field derived from the Gaussian distribution along the sketch – for multiple agents. In this particular scene, we are simulating 200 insects orbiting around a lantern. Similarly, taking sketch at the right side of Figure 7 as input, we generated the simulation of 50 butterflies moving on a prairie (Figure 5 Right). Our technique can also be easily integrated with existing global path planning algorithms. An example shown in Figure 6 demonstrates our framework incorporated with a navigation algorithm similar to the one presented in [PvdBC*11].

† The virtual environments' creation and design are inspired by Andrew Price.

5.2. Evaluation

For evaluating our technique, we conformed the size of our simulation domain to the same scale as the data capturing environment, and adopt both qualitative and quantitative measurements.

To start with, one of the most prominent features of swarming phenomenon is the spatial pattern and corresponding density information. The comparison result can be seen in Figure 4. As of random waypoints sampling and our mesoscopic waypoints sampling approaches, the latter tends to have small clusters reflected in intensity due to the fact that most processed segments have relatively small length comparing to the capturing environment. Thus by sampling the segment length distribution, agents travel frequently in short distances. After combining the macroscopic level sampling, the density distribution of our simulation approximates closely to the data and demonstrates the spherical pattern as of the data. In this case, the density decreases when the distance to swarm centroid increases resulting the aggregation gathered around the centroid. This finding is consistent with previous literature in biology and also dispels the assumption that insects can be simulated in a purely random fashion which would result in roughly uniform distribution within any predefined simulation space.

While visual similarity is an intuitive way to evaluate the results, for large dynamic systems, we also need quantitative measurements. In particular, as we are using Metropolis-Hastings algorithm to govern mesoscopic-level waypoints generation based on discrete density distribution constructed from the data, different discretizing scales of the simulation space could alter simulation qualities. In Figure 8, we show by altering cell numbers the change of average distance of all agents to the swarm centroid during a period of time. Notably, adding mesoscopic level computation increases the average distance and the reason is probably by traveling short distances and with random assigned initial locations, agents tend to stay closely to their initial positions while under random assigned waypoints agents travel across the whole space more frequently resulting shorter distances to the centroid at many timesteps.

Another notable feature of biological systems is the order in collective behaviors. While swarms of flies are rarely similar to other animal species (e.g., fish) that would polarize their state and make the group behave as a whole, they have been reported do show some features as collective behaviors [ACC*13]. To measure such an order, we use the polarization factor which is a common metric in studying the collective animal behavior [VCBJ*95]. The polarization Φ of a swarm is calculated as follows:

$$\Phi = \|\frac{1}{N}\sum_{i=1}^{N}\frac{\vec{v_i}}{\|v_i\|}\| \tag{8}$$

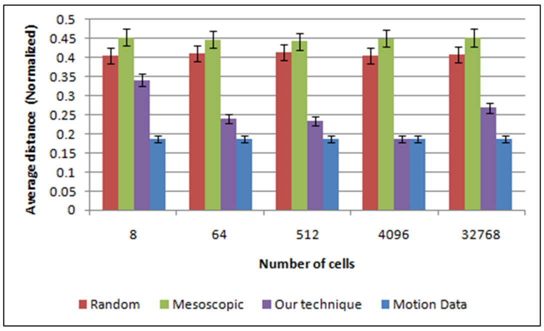

Figure 8: *Agents' average distances of different sampling techniques on various discretizing scales comparing to the motion data.*

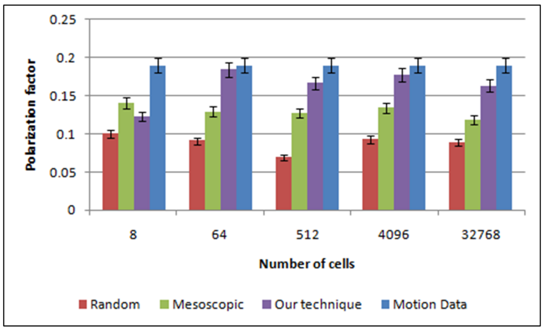

Figure 9: *Polarization scores of different sampling techniques on various discretizing scales comparing to the motion data.*

where $\vec{v_i}$ indicates the velocity of ith agent and N is the total number of agents. When Φ is zero, it means all individual velocities are canceling each other in terms of the direction. When Φ is close to one, it means velocities of all agents have nearly parallel directions. The values of polarization Φ at different discretizing scales under various sampling methods are shown in Figure 9. In general, by combining more waypoints sampling techniques we get closer Φ as compared to the reference data.

Worth noting, for both the average distance and polarization factor analyses, when the cell number gets either too small or too large, the Metropolis-Hasting algorithm appears to be less effective. This is due to the discretizing scale affects the visiting frequency of each cell and further influences the spread of the underlying density distribution. Extreme density spreads would cause either generated samples are far away (when spread is too large) or longer markov chains traverse the density resulting low probability regions undersampled (when spread is too small) [CG95].

For algorithmic analysis, the pre-processing time and memory complexities are $O(n)$ where n is number of frames and the sampling methods take $O(1)$ time. On a typical laptop (Intel i5-3230@2.60GHz, 6GB RAM, Windows 8.1 64 bits), the simulation of 100 agents, using ORCA, runs at 4,900 FPS.

6. Conclusion

We present a multi-scale, data-driven framework for visual simulation of insect swarms. We simulate each insect as an individual agent; at each stage of simulation we select the appropriate techniques or parameters based on the analysis of observed/captured insects motion data. Like many agent-based methods, our method consists a local avoidance module and a waypoint synthesis module to capture local interaction among insects and noisy individual trajectories that result in the coherent macroscopic motion patterns of a swarm.

We show how to automatically select each module of our method based on recorded/observed insect motion data. In general, it is very difficult for an animator to control a swarm motion, as it is nearly impossible to specify motion trajectory for each individual insect, simulation is the most efficient way to create insect swarms. However, the parameter space is large and a desired result can be achieved only after very time-consuming manual, trial-and-error process. Our algorithm avoids this tedious tuning task and reproduces the visual animation of insect swarming motion with ease. Our method is also user-friendly, as it enables direct manipulation of large-scale insect swarming motion by allowing the animator to sketch out a desirable path and incorporate this user input into density gradient maps without changing other modules. In addition, as our approach runs at very high interactive rates, animators can apply iterative refinement on simulation till the visual quality of animation results is satisfactory.

We tested, evaluated, and demonstrated our approach on several scenarios both qualitatively through visual inspection and quantitatively using scoring metrics. In particular, our quantitative comparisons illustrate the ability of our simulation to reproduce statistically significant measurements.

6.1. Limitations and Future Work

There are several possible existing future directions to extend our approach. First, it would be simpler to integrate our approach with 2D videos instead of 3D insects motion captured data. Our method requires insect swarm motion data, which provides the biological basis to our approach and enable automatic configuring of a simulation algorithm. The time-consuming tasks of selecting the appropriate collision avoidance technique and manually tuning simulation parameters are avoided. Although 3D motion capture technology is becoming more prevalent, recorded insect motion data are still difficult to obtain, while video recording can be easily done on commodity devices. Using 2D video input, however, raises several technical challenges, such as insect tracking in videos or 3D motion parameters reconstruction from 2D input. These are interesting computer vision problems by themselves.

Second, our current implementation assumes that local interaction among insects is largely driven by local collision avoidance and that each behaves according to the same statistical laws for simplicity and efficiency in simulation, as commonly done by many multi-agent simulations. In reality, there may be multiple types of insects interacting with each other and their social behaviors can vary. Clustering trajectories before processing them may introduce a new ability to consider mixed data from multiple species or more complex insects behaviors. Or, one can imagine different factors to consider and evaluate them for different types of insects (e.g. crawling vs. flying).

While we consider 3D motion data as a more general case for testing our algorithm, our method is perfectly able to handle 2D insect motion data. Adapting our data processing pipeline to such a situation is straightforward. However, due to the lack of 2D insects motion data, We have not tested our method on simulating (crawling) insects moving on a 2D plane. In addition, we consider a static density field to guide the global, aggregate insects motion based on the Metropolis-Hastings algorithm. It would be interesting to consider dynamic fields and to automatically adapt these fields to significant changes in the surrounding environment. Finally, the promising results of this paper suggest possibility of developing more novel tools for animators to guide simulated swarm motion for artistic control.

Acknowledgments

This work was supported in part by National Science Foundation, the SIMS Inria Associated Team, and the UNC Arts and Sciences Foundation. We are especially grateful to Dr. James Puckett at Gettysburg College and Dr. Nicholas Ouellette at Yale University for sharing their data with us. We would also like to thank Prof. Dinesh Manocha for introducing us to the data source and Xin Zhao for insightful discussions.

References

[ACC*13] ATTANASI A., CAVAGNA A., CASTELLO L. D., GIARDINA I., MELILLO S., PARISI L., POHL O., ROSSARO B., SHEN E., SILVESTRI E., VIALE M.: Collective behaviour without collective order in wild swarms of midges. *Cornell University Library arXiv:1307.5631* (2013). 3, 8

[BKBS13] BRECHT J., KOLOKOLNIKOV T., BERTOZZI A., SUN H.: Swarming on random graphs. *Journal of Statistical Physics 151*, 1-2 (2013), 150–173. 3

[BKHF14] BERSETH G., KAPADIA M., HAWORTH B., FALOUTSOS P.: Steerfit: Automated parameter fitting for steering algorithms. In *Proc. of the 2014 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2014), SCA '14, pp. 115–124. 4

[BMD*13] BUTAIL S., MANOUKIS N., DIALLO M., RIBEIRO

J. M. C., Paley D.: The dance of male anopheles gambiae in wild mating swarms. *J. Med. Entomol. 50*, 3 (2013), 552–559. 3

[BSC*06] Buhl J., Sumpter D. J. T., Couzin I. D., Hale J. J., Despland E., Miller E. R., Simpson S. J.: From disorder to order in marching locusts. *Science 312*, 5778 (2006), 1402–1406. 3

[CDF*03] Camazine S., Deneubourg J.-L., Franks N. R., Sneyd J., Theraulaz G., Bonabeau E.: *Self-Organization in Biological Systems*. Princeton University Press, 2003. 2, 6

[CG95] Chib S., Greenberg E.: Understanding the metropolis-hastings algorithm. *The American Statistician 49*, 4 (1995), 327–335. 6, 8

[CkJ*02] Couzin I. D., krause J., James R., Ruxton G. D., Franks N. R.: Collective memory and spatial sorting in animal groups. *J. Theory. Biol 218* (2002), 1–11. 3

[DCBC06] D'Orsogna M. R., Chuang Y. L., Bertozzi A. L., Chayes L. S.: Self-propelled particles with soft-core interactions: Patterns, stability, and collapse. *Phys. Rev. Lett. 96* (2006), 104302. 3

[FGLO99] Flierl G., GrÃijnbaum D., Levins S., Olson D.: From individuals to aggregations: the interplay between behavior and physics. *Journal of Theoretical Biology 196*, 4 (1999), 397–454. 3

[GCK*09] Guy S., Chhugani J., Kim C., Satish N., Lin M., Manocha D., Dubey P.: Clearpath: Highly parallel collision avoidance for multi-agent simulation. In *Proc. of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), SCA '09, pp. 177–187. 2

[HM95] Helbing D., Molnár P.: Social force model for pedestrian dynamics. *Physical Review E 51* (1995), 4282–4286. 2, 5

[Hug03] Hughes R.: The flow of human crowds. *Annual Review of Fluid Mechanics 35*, 1 (2003), 169–182. 2

[KLLT08] Kwon T., Lee K. H., Lee J., Takahashi S.: Group motion editing. In *Proc. of the 2008 ACM SIGGRAPH Conference* (2008), SIGGRAPH '08, pp. 1–8. 2

[KO13] Kelley D., Ouellette N.: Emergent dynamics of laboratory insect swarms. *Scientific Reports 3*, 1073 (2013). 3

[KS02] Kirchner A., Schadschneider A.: Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A 312*, 1-2 (2002), 260–276. 2

[MEK99] Mogilner A., Edelstein-Keshet L.: A non-local model for a swarm. *Journal of Mathematical Biology 38*, 6 (1999), 534–570. 3

[NGCL09] Narain R., Golas A., Curtis S., Lin M. C.: Aggregate dynamics for dense crowd simulation. In *Proc. of the 2009 ACM SIGGRAPH Asia Conference* (2009), SIGGRAPH Asia '09, pp. 122:1–122:8. 2

[NWK90] Neter J., Wasserman W., Kutner M.: *Applied Linear Statistical Models, 3^{rd} edition*. Burr Ridge, 1990. 3

[Oku86] Okubo A.: Dynamical aspects of animal grouping: Swarms, schools, flocks, and herds. *Adv. Biophys. 22* (1986), 1–94. 2, 3

[OPOD10] Ondřej J., Pettré J., Olivier A.-H., Donikian S.: A synthetic-vision based steering approach for crowd simulation. In *Proc. of the 2010 ACM SIGGRAPH Conference* (2010), SIGGRAPH '10, pp. 123:1–123:9. 2

[PAB07] Pelechano N., Allbeck J. M., Badler N. I.: Controlling individual agents in high-density crowd simulation. In *Proc. of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), SCA '07, pp. 99–108. 2

[PKO14] Puckett J., Kelley D., Ouellette N.: Searching for effective forces in laboratory insect swarms. *Scientific Reports 4*, 4766 (2014). 3, 6

[PvdBC*11] Patil S., van den Berg J., Curtis S., Lin M., Manocha D.: Directing crowd simulations using navigation fields. *Visualization and Computer Graphics, IEEE Transactions on 17*, 2 (2011), 244–254. 3, 7

[Rey87] Reynolds C. W.: Flocks, herds and schools: A distributed behavioral model. In *Proc. of the 1987 SIGGRAPH Conference* (1987), SIGGRAPH '87, pp. 25–34. 2, 5

[SESG08] Strefler J., Erdmann U., Schimansky-Geier L.: Swarming in three dimensions. *Phys. Rev. E 78*, 3 (2008), 031927. 3

[SKH*11] Singh S., Kapadia M., Hewlett B., Reinman G., Faloutsos P.: A modular framework for adaptive agent-based steering. In *Symposium on Interactive 3D Graphics and Games* (2011), I3D '11, pp. 141–150. 2

[Sum10] Sumpter D.: *Collective Animal behavior*. Princeton University Press, 2010. 2

[TBL06] Topaz C., Bertozzi A., Lewis M.: A nonlocal continuum model for biological aggregations. *Bulletin of Mathematical Biology 68*, 7 (2006), 1601–1623. 3

[TCP06] Treuille A., Cooper S., Popović Z.: Continuum crowds. In *Proc. of the 2006 ACM SIGGRAPH Conference* (2006), SIGGRAPH '06, pp. 1160–1168. 2

[TYK*09] Takahashi S., Yoshida K., Kwon T., Lee K. H., Lee J., Shin S. Y.: Spectral-based group formation control. *Computer Graphics Forum 28*, 2 (2009), 639–648. 3

[VCBJ*95] Vicsek T., Czirók A., Ben-Jacob E., Cohen I., Shochet O.: Novel type of phase transition in a system of self-driven particles. *Physical Review Letters 75*, 6 (1995), 1226–1229. 3, 8

[vdBGLM11] van den Berg J., Guy S., Lin M., Manocha D.: Reciprocal n-body collision avoidance. *Robotics Research 70*, 3-19 (2011), 260–276. 2, 5

[WGO*14] Wolinski D., Guy S., Olivier A.-H., Lin M., Manocha D., Pettré J.: Parameter estimation and comparative evaluation of crowd simulations. *Computer Graphics Forum 33*, 2 (2014), 303–312. 4, 5

[WJDZ14] Wang X., Jin X., Deng Z., Zhou L.: Inherent noise-aware insect swarm simulation. *Computer Graphics Forum* (2014). 3

[YEE*09] Yates C. A., Erban R., Escudero C., Couzin I. D., Buhl J., Kevrekidis I. G., Maini P. K., Sumpter D. J. T.: Inherent noise can facilitate coherence in collective swarm motion. *Proceedings of the National Academy of Sciences 106*, 14 (2009), 5464–5469. 3