Context-Aware Motion Diversification for Crowd Simulation

Qin Gu and Zhigang Deng ■ *University of Houston*

Crowd simulation models typically focus on navigational pathfinding and local collision avoidance; little research has explored how to control individual agents' motions. A proposed approach adaptively controls agents' motion styles to increase a crowd's visual variety. Experimental scenarios and user evaluations demonstrate the approach's flexibility and capability.

Large-scale crowd simulation is important in computer games, movies, virtual training, and education applications. One popular approach, agent-based crowd simulation, considers the properties of each individual (agent) separately at every time step, enabling a highly realistic simulation of path navigation, cognitive reaction, collision avoidance, and animation control.

Agent-based crowd simulation systems work according to a three-layer hierarchy. The highest layer provides the navigation waypoints through pathfinding and decision-making. The intermediate layer achieves collision avoidance and collision response by computing high-level motion information for every update (that is, every time step) using perceptual rules or social forces. The lowest layer handles the detailed animation of each agent according to the parameters supplied by the higher layers. For each agent, these three layers answer three questions:

- Where's the final target?
- Where's the next step?
- How should each agent perform its motion in the next step?

Although many researchers have proposed approaches for global navigation and local perception (the top two layers), relatively few have focused on controlling agents' detailed motions throughout a crowd—for example, how to efficiently control motion diversity among the agents.

In most crowd simulation models, many agents will be performing the same *motion type*—that is, a general category of motion such as walking, running, or waiting. To appear plausible, however, a motion type can't look identical for each agent performing it. The ideal way to achieve the necessary diversity is to assign a unique *motion style* (a variation of a motion type) to each agent during a certain time period, because individuals in real-world crowds have unique motion styles based on their distinctive personalities. However, such an approach incurs prohibitive computation and resource costs. So, the problem of improving a crowd's motion diversity (or variety) becomes how to make the crowd look plausibly diverse, given a limited number of available motion styles.

We've developed an approach that attempts to solve this problem. Basically, it maximizes the style variety of local neighbors and maximizes global style utilization while maintaining a consistent style for each agent that's as natural as possible. It only requires high-level motion information (such as speed and motion type) computed from the crowd simulation system's navigation and perception layers. As such, it can complement high-level crowd simulation models. Several experiment scenarios and a perceptual user study have demonstrated the flexibility and superiority of our approach over the traditional random distribution of motion styles.

 Published by the IEEE Computer Society

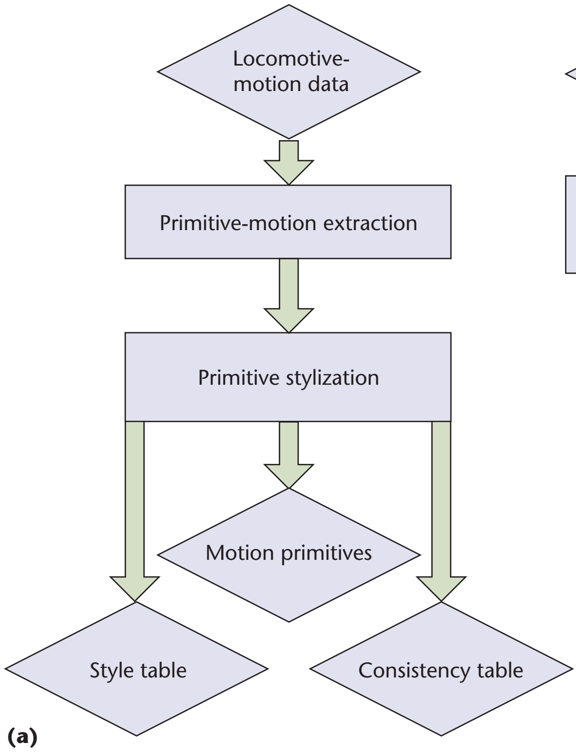


Figure 1. The pipeline overview of our approach. (a) Offline motion stylization. (b) Runtime motion diversity control. Precomputing crucial information in the offline stage makes the runtime motion diversity control highly efficient.

Pipeline Overview

Figure 1 shows the pipelines of our approach's two main components. During offline preprocessing (see Figure 1a), we segment and extract primitive human motions from a motion capture database, including cyclic walking and running motions and acyclic waiting and fighting motions. Then, a stylization process parameterizes and sorts the motions on the basis of kinetic energy. We next generate style variation tables and compact consistency tables for the runtime query. (We discuss these tables in more detail later.)

At runtime (see Figure 1b), the animation layer retrieves feature vectors from the higher layers of a crowd simulation system to decide each agent's motion type and velocity. Then, our novel motion diversity control selects proper motion styles for individual agents at each update.

Motion diversity control has three elements, based on three simulation premises. First, an agent's motion style should maximize the local style variety among its neighbors, so that the same or highly similar motion styles aren't clustered. Second, the motion style should maximize the overall diversity of motion styles in the crowd. Finally, for certain motions such as cyclic walking, the motion style should be as consistent as possible with the agent's current style to prevent unrealistic sharp changes of motion.

Offline Motion Preprocessing

To assist the runtime motion diversity control, we first generate primitive motions with associated style information. Although our motion diversity control is independent of motion type, with no loss of generality, here we consider only walking, running, fighting, and waiting motions because they're common in crowd simulations such as battlefields and urban streets.

We choose motion capture data because of its accuracy and realism. However, creating a large motion capture database for every crowd simulation project isn't always practical. So, we propose a data-driven method to extract and stylize primitive motions from a publicly available motion capture database.

Primitive-Motion Extraction

Over the past few years, several researchers have studied retrieving characteristic motions from a large motion database. Both semantically and numerically based retrieval approaches have achieved impressive accuracy in classifying motions.[1] Because large-scale crowd simulations require high performance, we need an efficient, concise feature vector to characterize motion styles.

Several researchers have tried to analyze, decompose, and quantify human motions. Nikolaus Troje presented an efficient framework for decomposing

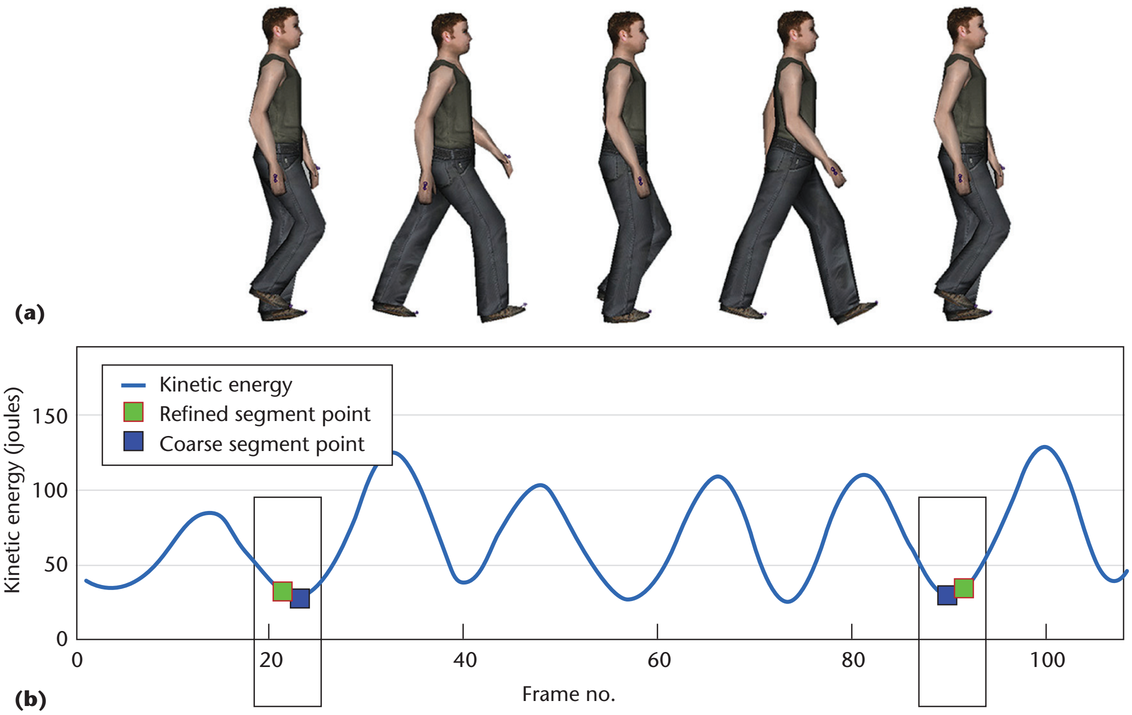


Figure 2. Primitive-motion extraction. (a) A full walking cycle. (b) The associated kinetic-energy segmentation (the refining window size is 7). This procedure provides a coarse segmentation for cyclic walking and running motions.

walking motions to a low-dimensional representation for analysis and synthesis.[2] However, he didn't establish whether you can soundly and robustly apply this framework to other types of human motions besides walking. Liu Ren and his colleagues explored statistical models to quantify the naturalness of various human motions.[3] But their approach focused on the qualitative judgment of natural versus unnatural aspects of human motions, so it can't produce a quantitative feature vector for characterizing each motion style.

Our research is inspired by the distance function that Kensuke Onuma and his colleagues proposed.[4] We compute the instant kinetic energy from the joint angular velocity and joint moment of inertia to retrieve primitive motions. We then adopt the mean kinetic energy to stylize each primitive. Compared with the original joint-angle-motion data, the instant kinetic energy doesn't vary with reciprocal limb motions. For instance, during walking, one leg's raw movement angle will neutralize the other leg's negative movement.

Proper segmentation is critical for cyclic walking and running motions. We retrieve primitive-motion segments that start from the single-foot contact state and consist of a full cycle of motions (see Figure 2). We base this retrieval on three observations:

- A full motion cycle's starting and ending poses should be as similar as possible to optimize runtime blending.
- The pose for switching between real-world walking, running, and standing is always the single-foot contact pose.
- The single-foot contact pose is more common than any on-the-fly poses, regardless of the motion style.

We can roughly identify primitive-motion segments by analyzing the entire human body's kinetic-energy trajectory, because a locomotive motion exhibits highly cyclic patterns. Figure 2 shows the low-pass filtered kinetic-energy curve of a walking motion. We use Onuma and his colleagues' method[4] to compute each joint's moment of inertia. Unlike computations using the center-of-mass trajectory, their method gives kinetic-energy values that fall into a local minimum on both constrained poses (such as the single-foot stage) and unconstrained poses (such as the double-foot stage for walking and the flying stage for running). This helps us unify a segmentation solution for both walking and running motions. We can segment a full locomotion cycle by starting from a single-foot contact stage (a local minimum and foot contact with ground) and searching for five consecutive local valleys. If we encounter a bursting spike (a sharp turn of motion) or a number of continuous near-zero values

Related Work in Visual Variety in Crowd Simulation

In recent years, agent-based crowd simulation models that rely on sophisticated global path planning and local collision dynamics for each crowd member have attracted increased attention. Among them, force-based models and their various extensions such as Nuria Pelechano and her colleagues developed[1] apply repulsion and tangential social forces to drive interactions between agents or subgroups. Following Craig Reynolds's seminal research on generating steering behaviors for flocks, herds, and schools,[2] rule-based crowd simulation has achieved highly realistic human behavior in complicated environments. In addition, the widely known motion-graph algorithm has found a role in retrieving and playing back the appropriate motion data in crowd simulations.[3]

Visual variety or diversity affects the overall perception of realism in many crowd simulation scenarios, such as a street with a high density of pedestrians. Owing to computation and resource limitations, most real-time simulation systems must repeat agent appearances or motion patterns for efficient performance, with a corresponding sacrifice of crowd diversity. Researchers have proposed several approaches for enriching the appearance variety among agents, such as recoloring textures for different body parts at runtime,[4] modulating illumination maps,[5] manipulating combinations of personal accessories, and scaling body skeletons for different body heights.[4] In addition, Gunnar Johansson has investigated visual perception of biological motions and found that a 10- to 12-moving-dot representation was adequate to evoke a compelling impression of human motions (such as locomotion).[6]

The research most related to our proposed approach is Rachel McDonnell and her colleagues' perception study of crowd variety.[7] That study produced several design implications and rules for crowd variety, such as how appearance clones, motion clones, and their combinations can affect the perceived variety of a crowd. McDonnell and other colleagues further evaluated the perceptual influences of different parts of a human body in a crowd.[8] These evaluations prove the effectiveness of adding appearance variety and illustrate that adding motion styles can also contribute to disguising clone effects. Compared with generating different agent appearances at the beginning that remain fixed during the simulation, dynamically changing agents' motion styles over time is somewhat less detectable, thus providing more controllability at runtime.

Researchers have also investigated how to generate motion variations based on a given character motion dataset. For example, Manfred Lau and his colleagues developed a dynamic Bayesian-network model to evaluate motion variations at high speed in both temporal and spatial domains.[9] However, directly applying these single-character animation techniques to a large-scale crowd with thousands of agents is too computationally costly for real-time applications.

References

1. N. Pelechano, J.M. Allbeck, and N.I. Badler, "Controlling Individual Agents in High-Density Crowd Simulation," *Proc. Siggraph/Eurographics Symp. Computer Animation* (SCA 07), Eurographics Assoc., 2007, pp. 99–108.
2. C.W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model," *Proc. Siggraph*, ACM Press, 1987, pp. 25–34.
3. L. Kovar, M. Gleicher, and F. Pighin, "Motion Graphs," *ACM Trans. Graphics*, vol. 21, no. 3, 2002, pp. 473–482.
4. J. Maïm, B. Yersin, and D. Thalmann, "Unique Character Instances for Crowds," *IEEE Computer Graphics and Applications*, vol. 29, no. 6, 2009, pp. 82–90.
5. S. Dobbyn et al., "Geopostors: A Real-Time Geometry/Impostor Crowd Rendering System," *Proc. ACM Siggraph Symp. Interactive 3D Graphics and Games* (I3D 05), ACM Press, 2005, pp. 95–102.
6. G. Johansson, "Visual Perception of Biological Motion and a Model for Its Analysis," *Attention, Perception, & Psychophysics*, vol. 14, no. 2, 1973, pp. 201–211.
7. R. McDonnell et al., "Clone Attack! Perception of Crowd Variety," *ACM Trans. Graphics*, vol. 27, no. 3, 2008, article 26.
8. R. McDonnell et al., "Eye-Catching Crowds: Saliency Based Selective Variation," *ACM Trans. Graphics*, vol. 28, no. 3, 2009, article 55.
9. M. Lau, Z. Bar-Joseph, and J. Kuffner, "Modeling Spatial and Temporal Variation in Motion Data," *ACM Trans. Graphics*, vol. 28, no. 5, 2009, article 171.

(a static pose) before the fifth local valley, we reset the current search.

This procedure provides a coarse segmentation for cyclic walking and running motions. To further ensure a seamless transition between extracted motion segments, we refine the coarse segmentation by applying a small window-based check around the first and last frame (see Figure 2). We compute the optimal (that is, the closest) frame pair as the final segmentation points using the metric that Lucas Kovar and his colleagues proposed.[5] (For more on this and other related research, see the sidebar.) Here, we experimentally set the check window's size to 7 for walking and 5 for running. The refining produces two types of motion cycles: one starting from the left foot and one starting from the right foot. In this article, we use only the right-foot cycle because we can easily swap the halves of the right-foot cycle to obtain the left-foot cycle.

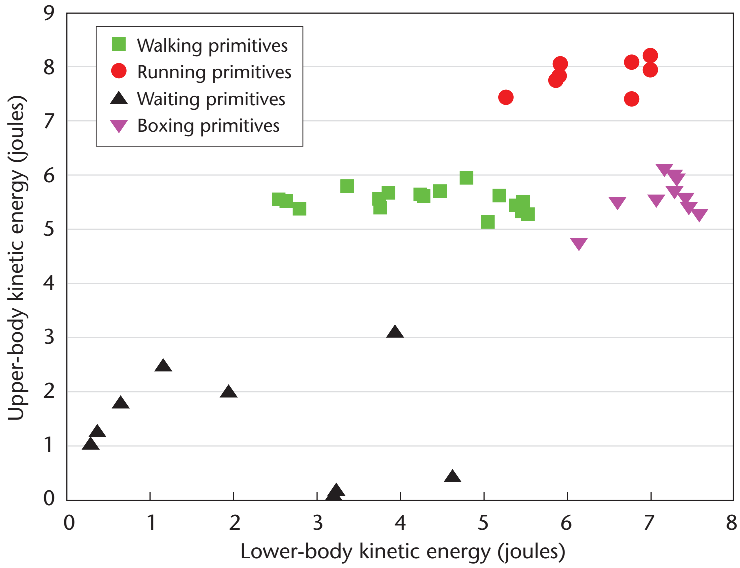


Figure 3. Feature vector distribution in the 2D lower-body and upper-body kinetic-energy space. The distribution should be roughly even to ensure that each style has at least one path to any other style in the consistency table.

Table 1. A style variation table for walking. The values indicate the Euclidean distance between two motion styles.

Motion style index number	Consistency ranking						
	1	2	3	4	5	6	7
1	0.00	1.73	3.46	5.20	6.93	8.66	10.39
2	1.73	0.00	1.73	3.46	5.20	6.93	8.66
3	3.46	1.73	0.00	1.73	3.46	5.20	6.93
4	5.20	3.46	1.73	0.00	1.73	3.46	5.20
5	6.93	5.20	3.46	1.73	0.00	1.73	3.46
6	8.66	6.93	5.20	3.46	1.73	0.00	1.73
7	10.39	8.66	6.93	5.20	3.46	1.73	0.00

Table 2. A consistency table showing the similarities between seven motion styles.

Motion style index number	Consistency ranking				
	1	2	3	4	5
1	1*	2	3	4	5
2	2	1	3	4	5
3	3	2	4	1	5
4	4	3	5	2	6
5	5	4	6	3	7
6	6	5	7	4	8
7	7	6	8	5	9

**To switch from motion style 1 to target style 7, the agent first moves from style 1 to style 5, and then from style 5 to style 7 at the next update.*

On the other hand, most acyclic motions, such as fighting and waiting, don't exhibit repeated motion patterns. So, we retrieve acyclic primitives by detecting a long period of foot contact with a threshold of kinetic-energy change. This simple solution has been sufficient to extract waiting and fighting motions with comparable style variations.

Motion Stylization

To control individual agents' motion diversity, we need a metric to quantitatively measure the difference between a pair of primitive motion styles. Our stylization process first categorizes unlabeled primitive motions into different types and then classifies motions of the same type into different styles.

The logarithm of the mean kinetic energy[4] is an effective metric for our purpose because it's independent of the motion's length. For motion clustering and classification, we empirically choose the following 2D feature vector, composed of the logarithm of the mean kinetic energy of the upper and lower body parts:

$$(\log(E_{\text{upperbody}} + 1), \log(E_{\text{lowerbody}} + 1)).$$

Within each motion type, this metric also quantifies the style variations. In our experiments, this metric generally produced a perceptually sound motion style ranking. Figure 3 shows the 2D distribution of the primitive motions' stylization metrics.

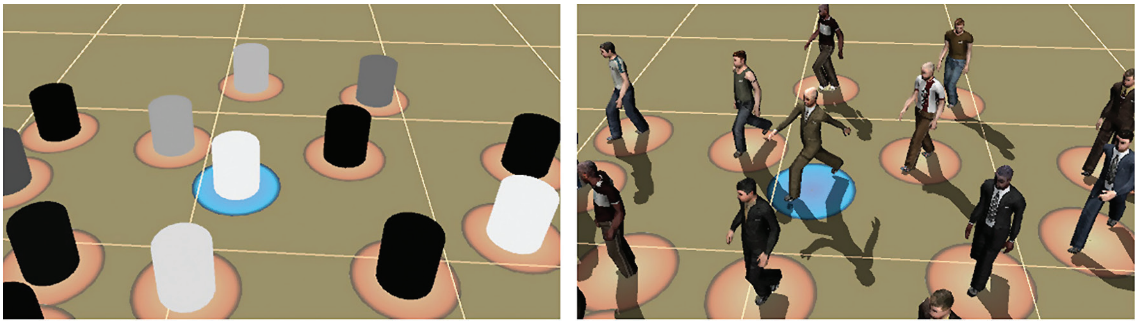
To accelerate runtime motion selection, for each motion type we also generate a style variation table (see Table 1) of size $s \times s$, where s is the number of styles for a certain motion type and each cell's value is the Euclidean distance between two motion styles. This data structure will serve as a lookup table for maximizing local motion diversity.

Furthermore, we generate a consistency table (see Table 2) that preregisters a small number of highly similar primitive motions for a specific style. We later use these as possible candidates in style selection. Keeping the number of motion styles in the consistency table small speeds up the search and selection process.

To prevent foot sliding, at runtime we must align the primitive motions' original speed with the speed computed from the crowd simulation system's high-level layers. We calculate a primitive motion's original speed by averaging the horizontal speed of its root, and we compute its runtime resampling factor as the ratio between the two speeds.

Dynamic Motion Diversity Control

A crowd's perceived diversity or variety relies on both appearance variety and motion variety. If many agents in a crowd have the same appearance or motion, the simulation will seem unrealistic. A simulation can achieve appearance variety by generating multiple 2D textures for the same 3D model. However, research has shown that adding



(a) **(b)**

Figure 4. The selected agent (blue) tries to perform the walking style that differs most from its neighbors (light orange). (a) A kinetic-energy representation of motion styles. (b) The corresponding 3D agent animations. To reduce the task's complexity, we register each agent into a discretized 2D grid at the beginning of every update and then look for only the nearest neighbors in the current agent's grid and the eight adjacent grids.

appearance variety doesn't increase the diversity of motion styles.[6]

Our motion diversity control explores an intelligent way to dynamically distribute limited motion styles across a large crowd. Given a specific motion type from the high-level crowd simulation modules, for an agent p at time t, we compute the optimal motion style for each next time interval, $S_p[t + 1]$, as the weighted combination of a local diversity function $D_p(S)$, a global utilization function $U(S)$, and a consistency management function $C_p(S)$:

$$S_p[t+1] = \arg\max_{S \subset R} \left(D_p(S) w_d + U(S) w_u + C_p(S) w_c \right). \tag{1}$$

Here, S is a motion style candidate from the space R of all available styles of the expected motion type. The values w_d, w_u, and w_c are user-defined parameters that weight different control components. The style-updating interval is the length of each primitive motion.

Maximizing Local Motion Diversity

The maximization of local motion diversity is inspired by the optimal graph-coloring problem. However, computing an optimal k-coloring for a set of nodes is NP-hard. Also, in a high-density crowd, there might be more neighboring nodes than available motion styles. So, we refine the selection criterion so that it finds the motion style most different from the styles of the neighboring agents in the local field of interest.

For example, in Figure 4a, the agents' grayscale levels indicate the kinetic-energy representations of their motion styles. Figure 4b shows the corresponding animations.

We compute $D_p(S)$ as

$$D_p(S) = \frac{1}{m} \sum_{q=1}^{m} \frac{StyleTable\left(S_q[t], S\right)}{NormalizedDistance_{pq}}.$$

Here, m is the number of neighbors around p; the numerator denotes the style difference between S and the current style of a neighbor agent q at time t, found in the style variation table. The $NormalizedDistance_{pq}$ between p and q gives more weight (importance) to a closer neighbor for style selection and less weight to a more distant neighbor.

We can't directly use the Euclidean distance between two agents in the world coordinate space because we compute $StyleTable(S_q[t], S)$ in a stylization-metric space. To avoid ending up with too large or too small a value of $D_p(S)$, we normalize the distance in the world coordinate space by the average distance between p and its m neighbors:

$$NormalizedDistance_{pq} = \frac{distance(pq)}{\frac{1}{m} \sum_{i=1}^{m} distance(pq_i)}.$$

Finding the m nearest neighbors is a bottleneck for both high-level perception simulation layers and our local diversity maximization. This bottleneck is due partly to the task's $\Theta(n^2)$ complexity, where n is the total number of agents in the simulation environment. We address this issue by registering each agent into a discretized 2D grid at the beginning of every update and then looking for only the nearest neighbors in the current agent's grid and the eight adjacent grids (see Figure 4).

The high-level perception layer typically considers only the neighbors in front of the agent, to mimic the restricted vision angle of humans (that is, agents). Our local diversity control, however, considers the neighbors from all directions because we want to disguise style clones from an audience instead of from other agents. Moreover, we take into account only neighbors with the same motion type because we generate different style variation tables for each motion type.

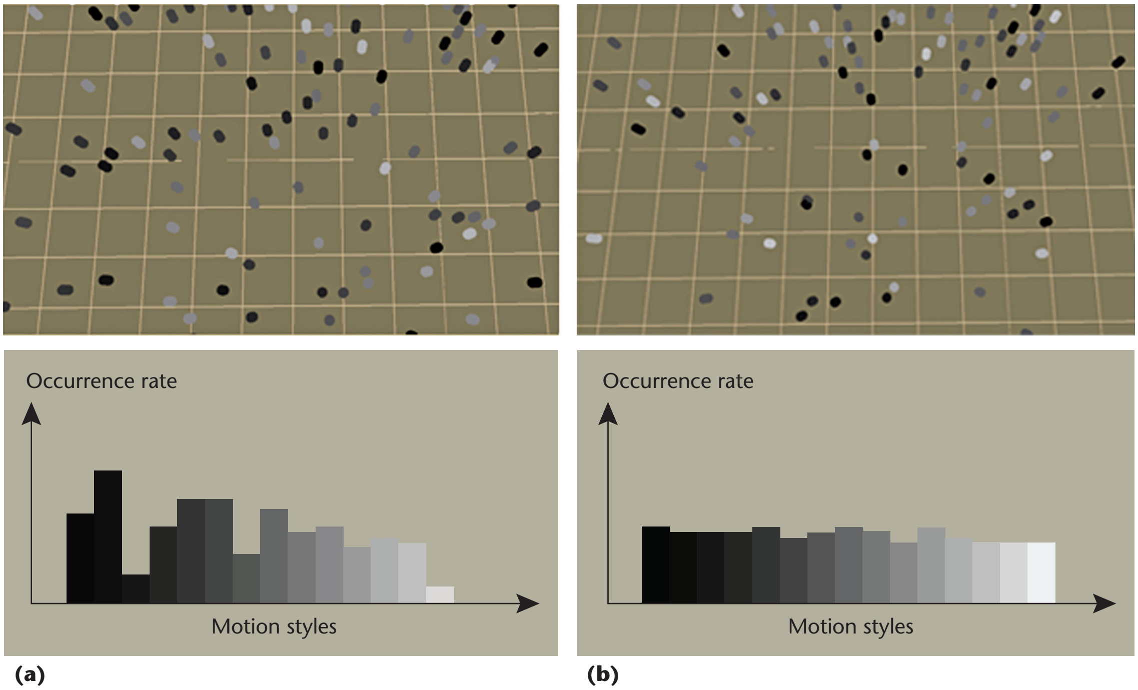


Figure 5. Global utilization control. (a) Random distribution of motion styles. (b) Our global utilization maximization. Our approach produces a near-uniform style utilization.

Global Utilization Control

We aim to maximize the use of every available motion style to achieve an approximately uniform distribution. The selected motion style contributes to the optimization of the global distribution of all available styles. For S, we determine the corresponding global utilization function $U(S)$:

$$U(S) = targetNum(S) - currentNum(S).$$

Here, $targetNum(S)$ is the expected number of occurrences of motion style S, and $currentNum(S)$ is the actual number of occurrences of S. $U(S)$ can be positive or negative. A crowd with only a few replications of S will produce a larger value than one with many examples of S already. A negative value indicates that S has already been "overcloned" and thus will be repulsed by our diversity control model.

It's simple to obtain $currentNum(S)$ by keeping a style counter over time. The following equation shows how we derive $targetNum(S)$ from the global distribution of styles:

$$targetNum(S) = P_S \times \frac{agentNum(T)}{styleNum(T)}. \quad (2)$$

Here, $agentNum(T)/styleNum(\mathrm{T})$ is the ratio of the number of agents with motion type T to the number of available styles for that motion type (assuming S is one of that type's motion styles); it represents the average style distribution across all agents. P_S denotes the priority of S; by default, the value is 1 for any S.

Figure 5 compares the results of a 200-agent crowd with and without global utilization maximization. A crowd simulation with a random distribution of motion styles often leads to an unbalanced style distribution, whereas our approach produces a near-uniform style utilization. Some scenarios call for a preference for certain styles over others, which means increasing their priority values in Equation 2.

Consistency Management

We also assume that an agent in a crowd should maintain its motion style as much as possible—if agents frequently switch among very different motion styles, the entire crowd will appear unrealistic and not smooth. Drawing on the style variation tables, we compute the consistency management function $C_p(S)$ for an agent p:

$$C_p(S) = \alpha_p \times \left(maxDistance - StyleTable\left(S_p[t], S\right)\right). \quad (3)$$

Here, $StyleTable(S_p[t], S)$ represents the difference between the current style of p and S, $maxDistance$ is the maximum value in the table for the difference between styles of the specific motion type that S belongs to, and α represents whether the expected motion type is the same as the current motion type. We consider only the consistency between two styles of the same motion type because changes in motion type are much more obvious and are normally controlled by the simulation's higher layers.

To maintain motion smoothness while maximizing crowd variety, we adopt the widely used level-of-detail concept. That is, we assume that agents closer to the viewing camera will attract more attention from an audience. Specifically, we apply the consistency management function $C_p(S)$ to only the agents in the viewing camera's range. For such an agent, the consistency management function's

Figure 6. Flocking behavior using our diversity control. Agents' motion styles converged to a stable stage after several updates.

weight is inversely proportional to the distance between the agent and the viewing camera:

$$\alpha_p = \begin{cases} 0 & \text{if } T_p[t+1] \neq T_p[t] \\ & \text{or Agent } p \text{ is out of view,} \\ \dfrac{threshold}{dist(p)} & \text{otherwise} \end{cases}$$

where *threshold* is a scaling parameter.

Remember that $D_p(S)$, $U(S)$, and $C_p(S)$ in Equation 1 were functions of S, which means that to find the optimal style, we would need to traverse the entire style space R of the same motion type. This is why we build a consistency table to reduce the size of R, on the basis of the following observations:

- With large numbers of styles and agents, searching the entire range of styles at every update is inefficient.
- Given the consistency constraint, our approach rarely selects candidate styles very different from the current style.

For each motion style in the style variation tables, we first sort other styles of the same motion type by kinetic-energy distance in ascending order and store only the indices of the first r styles in the consistency table (we set $r = 5$ empirically). For any motion style, the stored candidates are therefore the r most similar styles (those with a little higher or lower kinetic energy than the current style), starting with the current style itself (zero variation). Table 2 shows such a consistency table.

At runtime, an agent searches only the closest r candidates in the consistency table at each update for potential style transition targets. If users want to apply more than r different styles to an agent, the agent should choose the rth closest candidate style at the current update and then a more distant style in the next update. For example, considering $r = 5$ for Table 2, to switch from motion style 1 to target style 7, the agent first moves from style 1 to style 5, and then from style 5 to style 7 at the next update.

We also require that motion styles have a roughly even distribution in the stylization space (see Figure 3) to ensure that each style has at least one path to any other style in the consistency table. So, we remove redundant primitive motions (too close to or distant from other primitive motions) before generating the style variation table. Using the consistency table, we can empirically set *maxDistance* in Equation 3 to the rth closest candidate distance.

Results and Evaluation

We applied our approach to crowd scenarios generated by high-level crowd simulation models.[7,8] We also extracted 15 walking, 10 running, 10 fighting, and 10 waiting primitives from 11,138 frames of motion sequences from the Carnegie Mellon University Graphics Lab Motion Capture Database (http://mocap.cs.cmu.edu).

Experimental Scenarios

We chose four scenarios to showcase the influence of weight tuning in our motion diversity control.

The flocking scenario simulated typical crowd behavior; flock mates displayed similar locomotion patterns and targets (see Figure 6). We adopted the Boids model, wherein each agent moves according

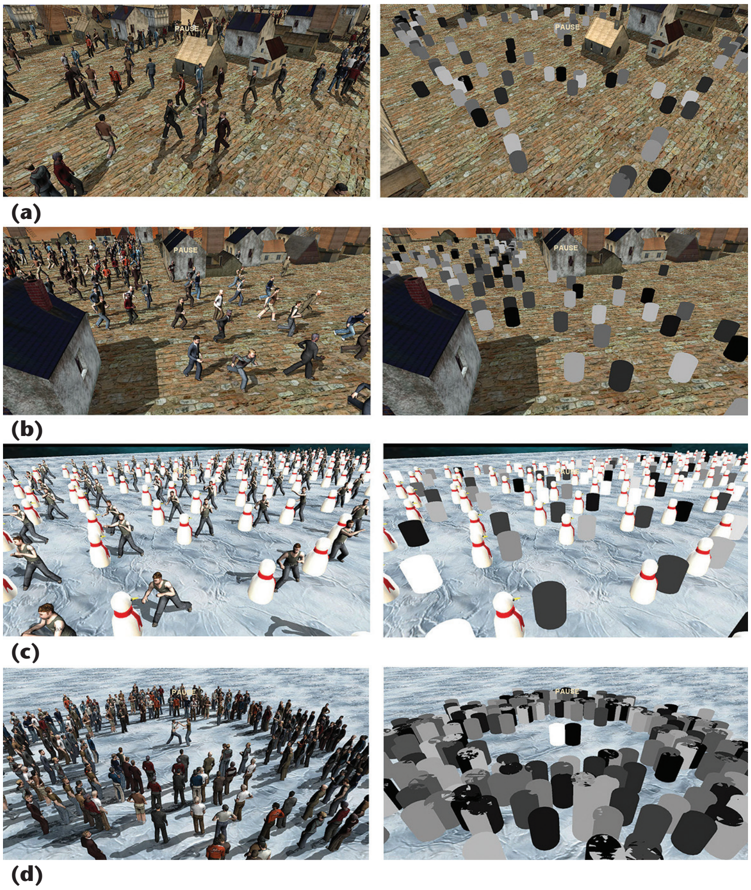


Figure 7. Our motion diversity control applied to scenarios simulated by the HiDAC (High-Density Autonomous Crowds) model. (a) An urban street. (b) Panicked evacuation. (c) Fighting training. (d) Street entertainment. As shown in this figure, our approach is able to simulate various crowd scenarios realistically.

to three steering behaviors: *separation*, *alignment*, and *cohesion*.[8] Audiences will easily view frequent switching among walking styles in this model as an unnatural effect because relative speeds and orientations among agents are consistent. So, in this scenario we gave motion style consistency higher priority than local diversity and global utilization control. This model's style distribution converged to a stable stage after several updates.

The crowded-town scenario tested types of cyclic motions using the HiDAC (High-Density Autonomous Crowds) model that Nuria Pelechano and her colleagues proposed.[7] The walking crowd (see Figure 7a) found the balance among the three control terms in Equation 2. Frequent switching between running styles in a panic situation (see Figure 7b) didn't produce obvious annoying effects. On the basis of these observations, we conclude that motion style consistency has a relatively low influence on the perception of high-frequency cyclic motions. This means the model can give heavier weights to local diversity maximization and global utilization control.

The frozen-land scenario showed how our approach applies to acyclic fighting, waiting, and watching motions (see Figures 7c and 7d). Unlike cyclic motions, acyclic motions usually benefit from a heavier weight for local diversity and global utilization, and a lighter weight for consistency management to achieve nonrepetitive motion patterns.

Finally, the military-march scenario showed our method's flexibility through locally manipulating the global utilization control. Whereas the default global utilization control tends to unify the style distribution, we can cluster specific motion styles by increasing their priority values P_S in Equation 2 for certain selected agents. This is particularly useful in simulating crowds moving in formation, as in a march. (See the video at http://doi.ieeecomputersociety.org/10.1109/MCG.2010.38 for the military-march simulation results.)

Table 3 presents the weight parameters (see Equation 1) used in these scenarios.

Complexity and Performance

Performance is a critical issue for agent-based crowd simulation systems because they must update each agent at every time step. An unoptimized version of our motion diversity control has a computational complexity of $n^2 \times s$, where n^2 is the

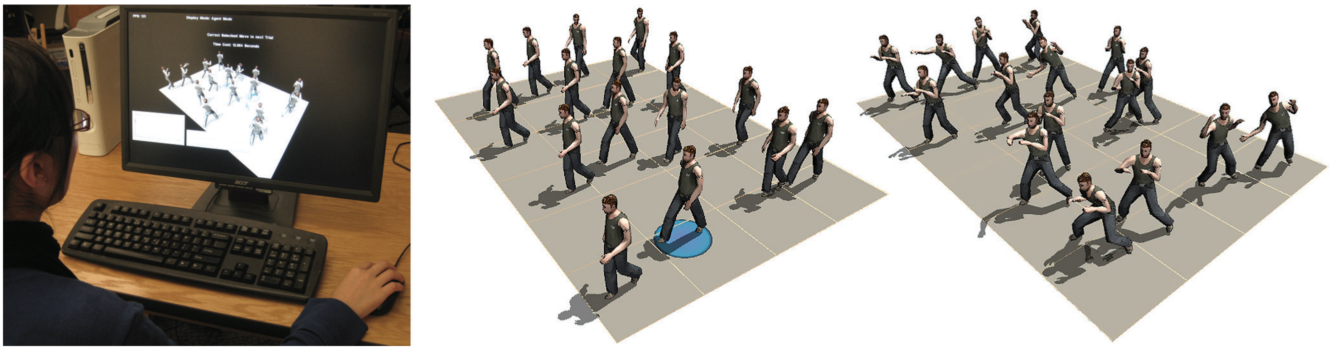
Table 3. Weight parameters for local diversity maximization, global utilization, and consistency management in Equation 1.

Scenario	Local diversity (w_d)	Global utilization (w_u)	Consistency (w_c)	Style priority (P_S)
Flocking	1.0	1.0	5.0	1.0
Crowded town—urban street	1.0	1.0	1.0	1.0
Crowded town—panicked evacuation	5.0	5.0	1.0	1.0
Frozen land—fighting training	10.0	5.0	1.0	1.0
Frozen land—street entertainment	10.0	5.0	3.0	1.0
Military march	1.0	10.0	5.0	10.0*

**In the military-march scenario, P_S is 10.0 only for marching walking styles.*

Table 4. Performance statistics.

Crowd size (no. of agents)	FPS for random distribution of motion styles	FPS for our approach	Computation overhead for our approach (%)
100	123.5	122.9	0.7
200	97.7	97.2	0.5
300	60.1	59.8	0.5
400	39.4	39.0	1.0



(a) **(b)** **(c)**

Figure 8. The perceptual user study. (a) The experimental interface. (b) A cyclic walking motion. (c) An acyclic fighting motion. For each trial, we uniformly generated the positions of 16 agents plus a small random offset.

cost of finding neighbors shared by higher-level perception layers and our diversity control and s is the number of motion styles. As we mentioned before, our optimized implementation reduces n^2 to n by registering agents into a discretized 2D grid at each frame. The consistency table decreases s to a small constant r, as we described earlier.

We tested our approach on an off-the-shelf PC with a 2.4-GHz CPU, 2 Gbytes of memory, and an Nvidia GeForce 260 graphics card. Using articulated 3D human models (800 to 1,000 polygons each) driven by high-quality motion capture data with 30 joints (62 degrees of freedom), we could simulate up to 500 agents at 30 fps. To test the computation overhead, we compared the average fps for the random distribution of motion style (the complexity was $\Theta(1)$) and our approach. The high-level simulation employed the HiDAC model with a discrete grid resolution of 50×50. Table 4 shows that our motion diversity control added only a small overhead.

Perceptual Evaluation

Numerically evaluating a simulated crowd is complicated because it depends highly on users' subjective perception. Rachel McDonnell and her colleagues provided an in-depth study on multiple factors that might affect the detection time of cloned motions from a crowd, including appearance, gait style, and the number of clones.[6] They used a random distribution of the cloned motions among the crowd without a control scheme. So, the usability question for our experiment is, does our approach make motion clones harder to detect than a random style distribution, using the same number of motion styles?

To answer this question, we showed basic simulations to 14 naive participants (12 male and 2 female). Most participants had little crowd simulation background. To minimize the influence of other simulation layers (for example, the navigation and perception levels), we fixed the positions and orientation of 16 testing agents with the same appearance. The experiment used the cyclic walking and acyclic fighting primitive motions in the scenarios we described previously, chosen from the 15 walking and 10 fighting motions. All agents faced the same direction and didn't collide with each other (see Figure 8).

Rather than using a single clone repeatedly, we tried to simulate more specific crowd situations that would allow multiple style clones multiple times. That is, given a limited number of motion styles, either random distribution or our approach determined which motion style to apply where. For the four trials, the available styles' upper limits were 2, 4, 6, 8, and 10. That meant 20 trials for each participant: 2 motion types $*$ 5 style pools $*$ 2 diversity control approaches.

Because finding all the clone pairs in a reasonable time frame would be difficult, we asked each participant to pick out one pair as quickly as possible. To mitigate the influences of different motion styles (some styles appear harder to identify) and fatigue, we counterbalanced the order of trials for the participants.

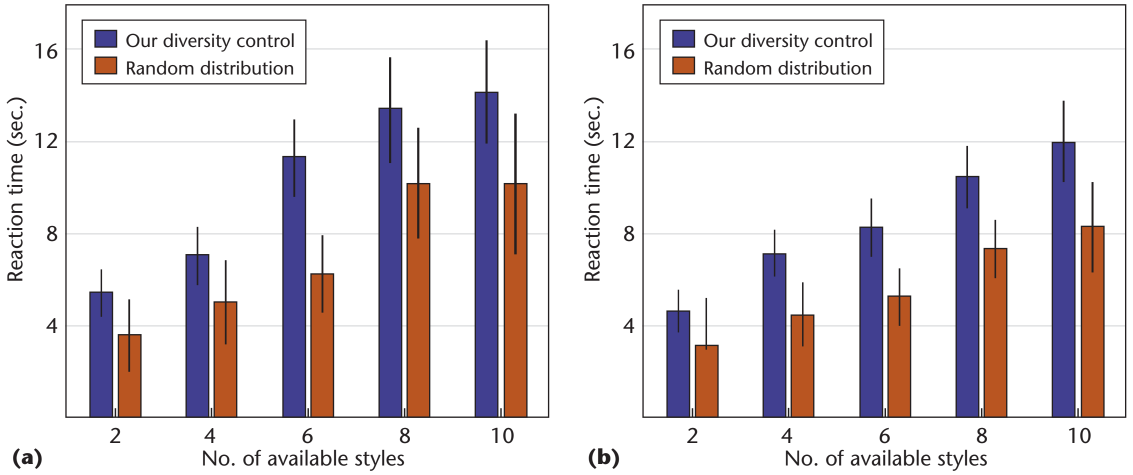


Figure 9. The average response time and standard deviation for detecting the first pair of motion style clones: (a) a cyclic walking motion and (b) an acyclic fighting motion. Our motion diversity control disguises motion clones more effectively than random distribution does, given the same number of available motion styles.

We used two-way analysis of variance to analyze the time participants took to pick the first clone style in each trial. Both the number of available styles (cyclic motion: $F = 26.25$, $p < 0.0039$; acyclic motion: $F = 38.40$, $p < 0.0019$) and the diversity control approach (cyclic motion: $F = 29.94$, $p < 0.0054$; acyclic motion: $F = 48.75$, $p < 0.0016$) are the main factors, with no evident interaction between them. The first result (the factor of the number of available styles) was consistent with the results other researchers have reported.[6] The second result, together with the average reaction time in both motion type conditions (see Figure 9), shows that our motion diversity control disguises motion clones more effectively than random distribution does, given the same number of available motion styles.

We didn't evaluate agent orientation's impact on clone detection, and we let the participants freely rotate the view. We observed that most participants preferred a side view (see Figure 8) instead of a front view to identify motion clones, contrary to what previous research found.[6] One possible explanation is that most participants tried to identify different styles through the swing magnitude of limb motions, which is easier to discern in a side view.

Our approach has several limitations. Currently, we don't use the transitions in the original motion capture data for two reasons. First, many interstyle transitions aren't available in the data. Second, pregenerating all the possible motion transitions among all styles demands nontrivial extra overhead for a large crowd. Because of performance concerns, we apply a spherical linear interpolation on agents' joint rotations and a linear interpolation on agents' translations to dynamically generate transitions at runtime. Because our consistency management mildly restricts the style change, we found the dynamically generated transition results visually acceptable. However, if a particular agent requires strict motion continuity—such as a computer game's main character—a more sophisticated motion synthesis method would be necessary.

Also, we use the average speed computed from the original primitive motions as the reference for computing the runtime animation resampling rate. This might still produce minor foot sliding for certain motions because the speed of real-world humans involves constant acceleration and deceleration, whereas the speed in high-level crowd simulation layers is typically constant.

In addition, the current algorithm considers every agent as the same type of person without variations in gender, personality, height, weight, or age. Recent research has indicated that the body shapes and even the motions of particular body parts will significantly influence a simulated crowd's overall visual variety.[9,10] We plan to explore more sophisticated motion style selection rules to account for these factors to further enhance visual realism.

As is common with data-driven methods, our approach's simulation results are limited to the motion database's capacity and variety. For example, if we use an extremely low number of available motion styles, viewers will easily detect motion clones. We plan to develop algorithms to synthesize motion style variations on the fly on the basis of current optimal motion selection outcomes, balancing visual realism and runtime performance.

Although our approach is independent of specific motion types, the offline stylization directly affects the final simulation results. Inappropriate stylization might cause jaggy effects in terms of consistency management, as when two styles have similar stylization values but aren't visually similar. Our segmentation and stylization processes generated sound results for the selected motions in this article. However, stylizing more complex human motions using compact feature vectors needs further exploration.

Our user study focused on how to effectively disguise motion style clones to increase perceived variety. An interesting future direction would be to investigate the effect of the change of motion styles in a crowd. This visual perception might vary with particular motion styles, agent distances, and numbers of agents. The findings would give additional insights to crowd motion diversity control.

Acknowledgments

This research was supported partly by US National Science Foundation (NSF) grants IIS-0914965 and CNS-0932272 and Texas Norman Hackerman Advanced Research Program grant NHARP 003652-0058-2007. Any opinions, findings, and conclusions or recommendations expressed in this article are the authors' and don't necessarily reflect the funding agencies' views.

References

1. Z. Deng, Q. Gu, and Q. Li, "Perceptually Consistent Example-Based Human Motion Retrieval," *Proc. ACM Siggraph Symp. Interactive 3D Graphics and Games* (I3D 09), ACM Press, 2009, pp. 191–198.
2. N.F. Troje, "Decomposing Biological Motion: A Framework for Analysis and Synthesis of Human Gait Patterns," *J. Vision*, vol. 2, no. 5, 2002, pp. 371–387.
3. L. Ren et al., "A Data-Driven Approach to Quantifying Natural Human Motion," *ACM Trans. Graphics*, vol. 24, no. 3, 2005, pp. 1090–1097.
4. K. Onuma, C. Faloutsos, and J.K. Hodgins, "FMDistance: A Fast and Effective Distance Function for Motion Capture Data," *Proc. Eurographics 2008—Short Papers*, Eurographics Assoc., 2008; http://graphics.cs.cmu.edu/papers/OnumaEG2008.pdf.
5. L. Kovar, M. Gleicher, and F. Pighin, "Motion Graphs," *ACM Trans. Graphics*, vol. 21, no. 3, 2002, pp. 473–482.
6. R. McDonnell et al., "Clone Attack! Perception of Crowd Variety," *ACM Trans. Graphics*, vol. 27, no. 3, 2008, article 26.
7. N. Pelechano, J.M. Allbeck, and N.I. Badler, "Controlling Individual Agents in High-Density Crowd Simulation," *Proc. Siggraph/Eurographics Symp. Computer Animation* (SCA 07), Eurographics Assoc., 2007, pp. 99–108.
8. C.W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model," *Proc. Siggraph*, ACM Press, 1987, pp. 25–34.
9. J. Maïm, B. Yersin, and D. Thalmann, "Unique Character Instances for Crowds," *IEEE Computer Graphics and Applications*, vol. 29, no. 6, 2009, pp. 82–90.
10. R. McDonnell et al., "Eye-Catching Crowds: Saliency Based Selective Variation," *ACM Trans. Graphics*, vol. 28, no. 3, 2009, article 55.

Qin Gu *is a PhD candidate in the University of Houston's Department of Computer Science. His research interests include computer graphics, character animation, and crowd animation. Gu has an MS in computer science from the University of Houston. Contact him at ericgu@cs.uh.edu.*

Zhigang Deng *is an assistant professor of computer science and the founding director of the Computer Graphics and Interactive Media Lab at the University of Houston. His research interests include computer graphics, computer animation, GPU computing, human-computer interaction, and medical data visualization and intervention. Deng has a PhD in computer science from the University of Southern California. He's a member of IEEE, ACM, and ACM Siggraph. Contact him at zdeng@cs.uh.edu or zdeng4@uh.edu.*