Crowd Simulation, Motion Planning (2)

CSE888.14X – Autumn 2008
Cheng Zhang

- Composite Agents (SCA 08)
  - By Yeh, H., Curtis, S., Patil, S., Berg, J. Manocha, D., and Lin, M.
- Group Motion Editing (SIGGRAPH 08)
  - By Taesoo Kwon, Kang Hoon Lee, Jehee Lee, Shigeo Takahashi

Composite Agents, SCA 08

Yeh, H., Curtis, S., Patil, S., Berg, J. Manocha, D., and Lin, M.

- Overview: an agent based approach with global road map for navigation and the reciprocal velocity obstacles for collision avoidance.
- A composite agent consists of a basic agent that is associated with one or more proxy agents.
- The idea is to inject intangible factors into the simulation by embodying them in "physical" form.

Contributions

- Introduce a simple and novel concept- *composite agents that* can easily model a variety of emergent behaviors for agent-based crowd simulation.
- The overhead of adding composite agents to the complex scenarios is negligible.

Definitions and Notations (1)

- Simulator – a general agent-based simulation system.
- *Agents* = $\{A_1, A_2, \ldots A_n\}$.
- Each agent has its own state, denoted as φ_i.
- This state can be categorized into an external state εi and an internal state i_i.
- εi represents properties of A_i that affect the motion of other agents in the system in computing collision-free paths.

Definitions and Notations (2)

- The internal state i_i include properties that are relevant to the agent itself but are not considered by other agents. Eg. the goal position, memory, or mental state.
- The environment φ_{Env} consists of the state necessary to navigate a collision-free path through the environment.

Composite agent formulation

- Composite agent formulation

$$
\begin{aligned}
proxy(A_i) &= \begin{cases} \emptyset & \text{for basic agents} \\ \{P_{i,1}, P_{i,2}, \ldots, P_{i,m}\} & \text{for composite agents} \end{cases} \\
parent(P_{i,j}) &= A_i.
\end{aligned}
$$

Proxy agent formulation

- A proxy agent's state includes:
  - External state – the same properties as the external state of basic agent;
  - Unique internal state.
- Can access its parent's internal state.
- Lead to the central idea behind composite agent – both basic agent and proxy agent are treated uniformly in Update().
- Update in p-Update() by a set of rules.

Simulation Algorithm

- for each $A_i \in Agents$
  - $Nbr \leftarrow \textsc{GatherNeighbors}(Agents_i^*)$
  - $\phi_i \leftarrow \textsc{Update}(\phi_i, E_{Nbr}, \Phi_{Env})$
- for each $P_{i,j} \in Proxies$
  - $\phi_{i,j} \leftarrow \textsc{P-Update}(\phi_{i,j}, \phi_i, \Phi_{Env})$

Influence of Composite Agents

- The fact that an agent reacts to both basic and proxy agents equivalently has a direct consequence.
- The influence that a composite agent C_i *exerts over other agents is extended beyond* its own external properties ε_i*, to indirectly include all* the influences of the $\varepsilon_{i,j}$*'s of its proxy agents.*

Influence of Composite Agents

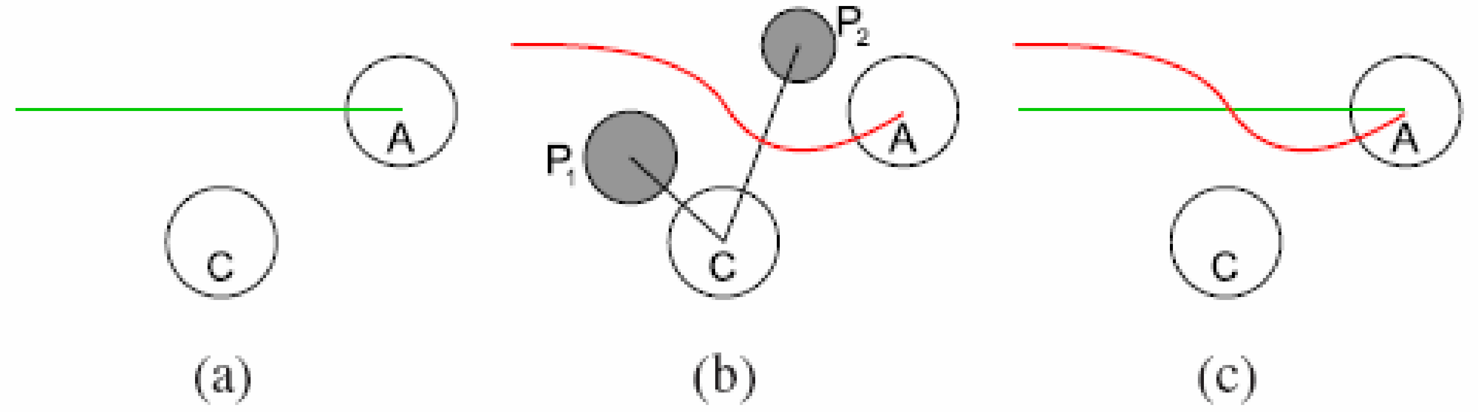


Figure 1: *Responses of an agent A encountering a composite agent C. (a)The green line shows the original planned path taken by A. (b) In the presence of the proxy agent of C, A takes the red path and avoids collision with P_1 and P_2. (c)Comparison of the paths.*

Modeling intangible factors

- Assumption: the external state consists of position, velocity and geometric representation.

$$\varepsilon = (\mathbf{p}, \mathbf{v}, \mathcal{G})$$

- Aggression, social priority, and authority.

Aggression (1)

- Aggressive behavior can be characterized as follows:
  - A person feels a sense of urgency—the desire to reach a goal more quickly.
  - The urgency is expressed in some manner causing other agents to either yield or steer clear.
- When a proxy agent $P_{i,1}$, *(aggression proxy)* is associated to the agent A_i. The proxy is placed near A_i in the direction it intends to move, as shown in Fig. 2(b).

Aggression (2)

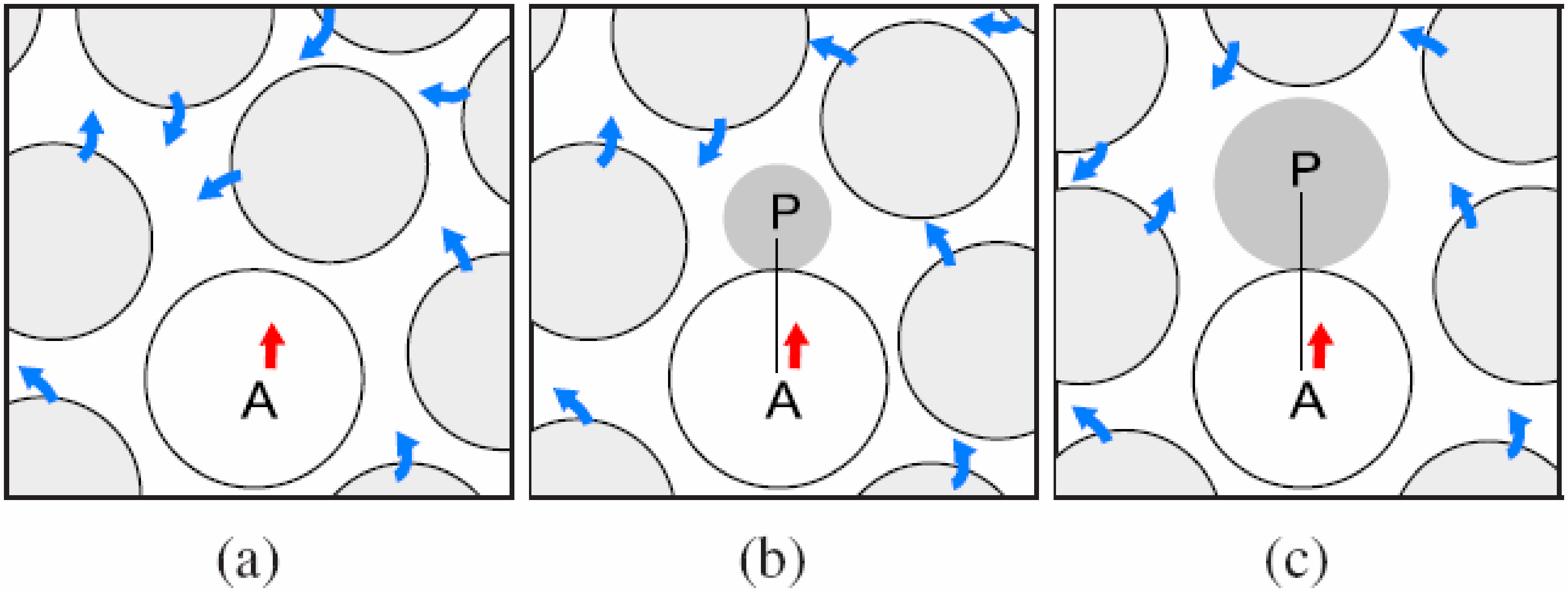


Figure 2: ***Aggression****: Agent A's desired direction is blocked. As A's urgency increases, its aggression proxy, P, grows and the other agents move to avoid it, leaving a space for A to move into.*

Aggression (3)

- Assume constant URGENCY
- P-Upudate()

- $\mathbf{p}_{i,1}$ is positioned at a distance from $\mathbf{p}_i$ in the direction that A_i intends to move.
- $\mathbf{v}_{i,1}$ is chosen to be identical to $\mathbf{v}_i$.
- $\mathcal{G}_{i,1}$ is a simple shape, such as a circle (as appropriate for the simulator.)

Aggression (4)

- Simulation with changing URGENCY
  - *Velocity-based urgency:* The greater the deviation of the current velocity from the preferred velocity, the greater this value grows.
  - *Distance-based urgency:* If the agent gets closer to the goal, the URGENCY value reduces; if the agent gets farther, the URGENCY value increases.

Aggression (5)

- p-Update() with changing URGENCY:

  - $\mathbf{p}_{i,1}$ is placed at a distance d, proportional to URGENCY, from $\mathbf{p}_i$;
  - $\mathcal{G}_{i,1}$ is scaled to a factor proportional to URGENCY, so that as A_i's urgency increases, so does the size of its proxy agent.

Social Priority (1)

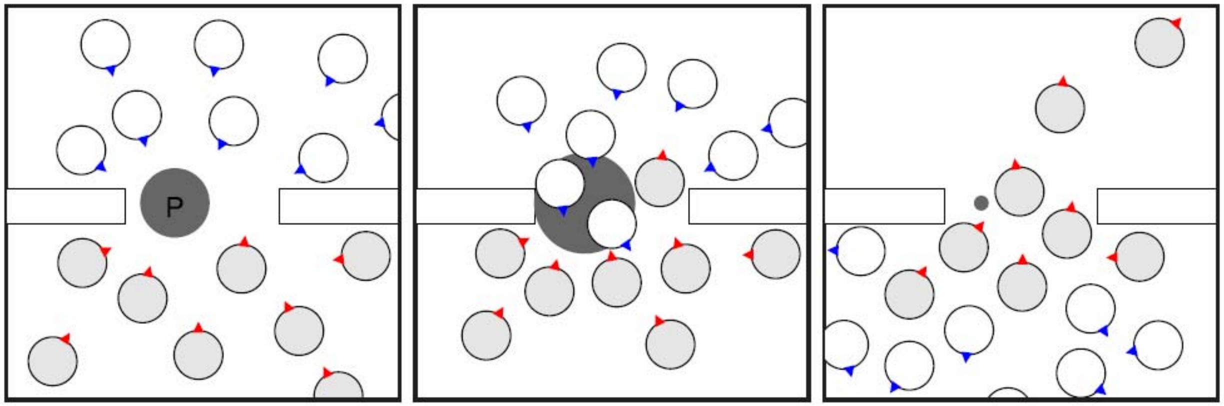
- New property: PRIORITY
- Set-up:
  - a basic agent has lower priority than all composite agents.
  - A proxy agent $P_{i,1}$, *(a priority proxy)* is placed at the contested location and grows as its parent A_i nears it.

Social Priority (2)

- p-Update():

- $\mathbf{p}_{i,1}$ is set right at the contested location;
- $\mathbf{v}_{i,1}$ is set to zero;
- $\mathcal{G}_{i,1}$ grows as A_i approaches the contested location, and shrinks as A_i leaves.

Social Priority (3)



(a) (b) (c)

Figure 3: ***Priority***: *The white agents should be given preference in passing through the doorway. (a) Each white agent has a priority proxy located at P and identical priority values. (b) As the white agents approach, the proxy grows, reserving the space for all of the white agents. (c) Finally, after the white agents have passed, the proxy shrinks to nothing and the gray agents may pass through unimpeded.*

Authority (1)

- Setup:
  - When a line of soldiers or policemen march into a dense crowd and they are still able to maintain a coherent line. Their authority makes it so that even if there is space between two consecutive members, civilians do not attempt to break the line.
  - Approximate this manifestation of authority with a *trailblazer,* who marks space that the members of his group can travel through while others cannot.

Authority (2)

- New property: TRAIL IDENTIFIER
- This property controls which "trail" a composite agent follows.
- When a set of proxy agents (*trail proxies)* is assigned to a composite agent, a trail proxy marks the path traveled by the composite agent.
- After every *T* seconds of the simulation, an agent places a proxy agent at its current position. The sequence of proxy agents marks the most recent segment of the path that the agent has traveled.

Authority (3)

- Consider a trailblazer A_i *and its proxy agents* $P_{i,1}, P_{i,2}, \ldots, P_{i,m}$.
- $P_{i,j}$ has a life cycle of period t that starts at time start j, and an age, represented as age j, which increases as simulation time passes. When age j becomes greater than t, age j is reset to 0, the starting time start j is set to the current time t, and the cycle starts again.
- At the beginning of the cycle, the position of the proxy is set to be that of the parent and its size is set to be the same as the parent. As the proxy agent ages, it shrinks.

Authority (4)

The P-UPDATE function for $P_{i,j}$ is expressed as:

- $\mathbf{p}_{i,j}$ is equal to $\mathbf{p}_i(start_j)$, i.e. where A_i was when the cycle started;
- $\mathbf{v}_{i,j}$ is zero;
- $\mathcal{G}_{i,j}$ is similar to $\mathcal{G}_i$ and scaled to the factor $1 - \frac{age_j}{\tau}$;
- internal state: age_j is increased by Δt; if $age_j \geq \tau$, age_j is set to 0 and $start_j$ is set equal to the current time t.

Authority (5)

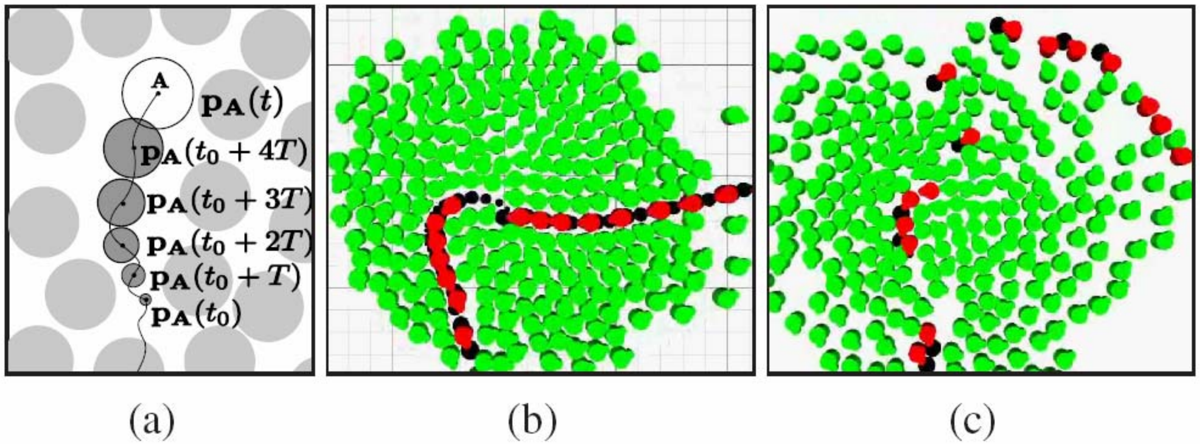


Figure 4: ***Authority***: *(a) An agent A and the trail (a sequence of trail proxies.) The trail proxies are placed at positions* $\mathbf{p}_i$, *at time instants* t_0, t_0+T, t_0+2T, t_0+3T *and* t_0+4T. *(b) A line of police maintains a formation while walking in a crowd; the police are associated with trail proxies and aggression proxies. (c) A simulation with the same initial configuration except without trail proxies.*

Protection/Guidance Behavior (1)

- Setup: when a child and his mother walk in a dense crowd, the mother protects the child from possible collisions and guides the child to stay on the current path.
- The mother needs to know where the child *K* is, predicts collisions for the child, and determines whether the child's moving direction is in a certain range;
- The mother needs to react to the situation, i.e. offering protection and guidance.

Protection/Guidance Behavior (2)

- Associate a proxy agent *P1 with the mother M*
- For protection behavior, suppose the mother detects that a stranger *S is*

- $\mathbf{p}_{i,1}$ is set to be in between K and S, say $\mathbf{p}_{i,1} = \frac{1}{2}(\mathbf{p}_k + \mathbf{p}_s)$;
- $\mathbf{v}_{i,1}$ is set to be equal to $\mathbf{v}_M$;
- $\mathcal{G}_{i,1}$: any shape that obstructs the trajectory for S to hit K.

Protection and Guidance Behavior (2)

- For guidance behavior, suppose the mother detects that *K is about to head outside of a region R, which she* thinks is an acceptable pathway, then

- $\mathbf{p}_{i,1}$ is set to be slightly outside of $\mathcal{R}$, along the line defined by $\mathbf{v}_k$;
- $\mathbf{v}_{i,1} = \mathbf{v}_M$;
- $\mathcal{G}_{i,1}$: any shape that is sufficient to block K.

Protection and Guidance Behavior (3)

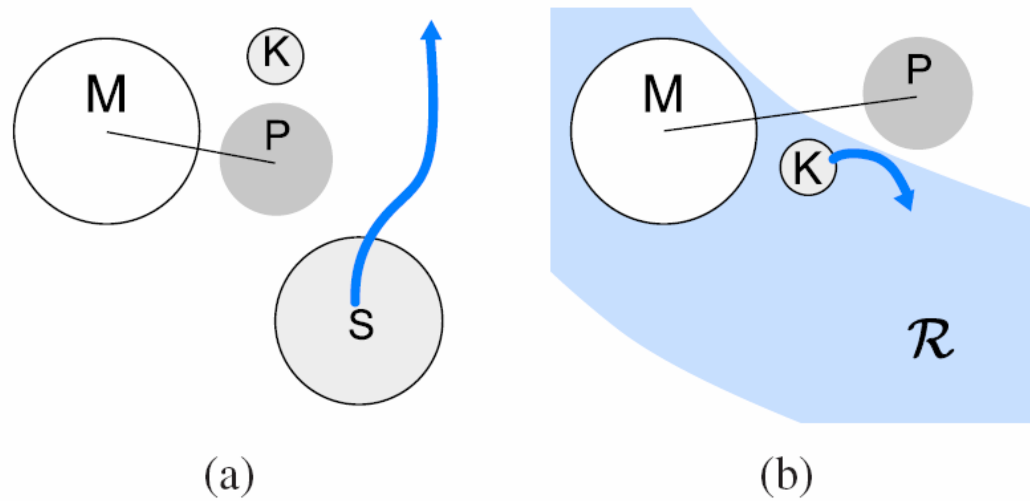


Figure 5: *(a) Protection: a mother M protects her child K by placing a proxy agent P in between K and an approaching stranger S. (b) Guidance: when K is about to stray from the correct pathway, indicated as the region $\mathcal{R}$, the mother places a proxy agent P just outside $\mathcal{R}$ to alter K's direction.*

Performance

Scene	#Basic agents	#Proxy agents	% Overhead simulation time	% Overhead memory usage	Type of proxy agents
Office	1000	47	1.9%	0.6%	aggression
Subway	340	100	0.3%	0.12%	priority
Embassy	240	200	10.75%	1.9%	trail, aggression

Table 1: *Performance of our approach on the three demo scenarios. The results indicate that the composite agent framework adds very little overhead to an existing multi-agent simulation system in terms of both simulation time and memory usage.*

Limitation

- Behaviors complicate communication or group coordination.
- Lack of precise control over the exact nature of the agent interactions due to the underlying planning system.

Group Motion Editing

SIGGRAPH 08

Taesoo Kwon, Kang Hoon Lee, Jehee Lee, Shigeo Takahashi

- A real-time approach to editing group motion as a whole while maintaining its neighborhood formation and individual moving trajectories in the original animation as much as possible.

Overview

- The graph structure - vertices represent positions of individuals at specific frames and edges encode neighborhood formations and moving trajectories.
- Editing operations
  - Deform group motion by pinning or dragging individuals.
  - Stitch or merge multi-group motions to form a longer or larger group motion while avoiding collisions.

Graph construction

- A graph G is characterized by the number of sampled frames T and the number of individuals N.
- For each frame, create a set of vertices $\{v_{i,j}\}N_{j=1}$, where vertex $v_{i,j}$ defines the two-dimensional location of individual j at frame i.
- The formation edges connect the vertices of a graph at each plane, and the motion edges connect two adjacent planes.

Graph construction

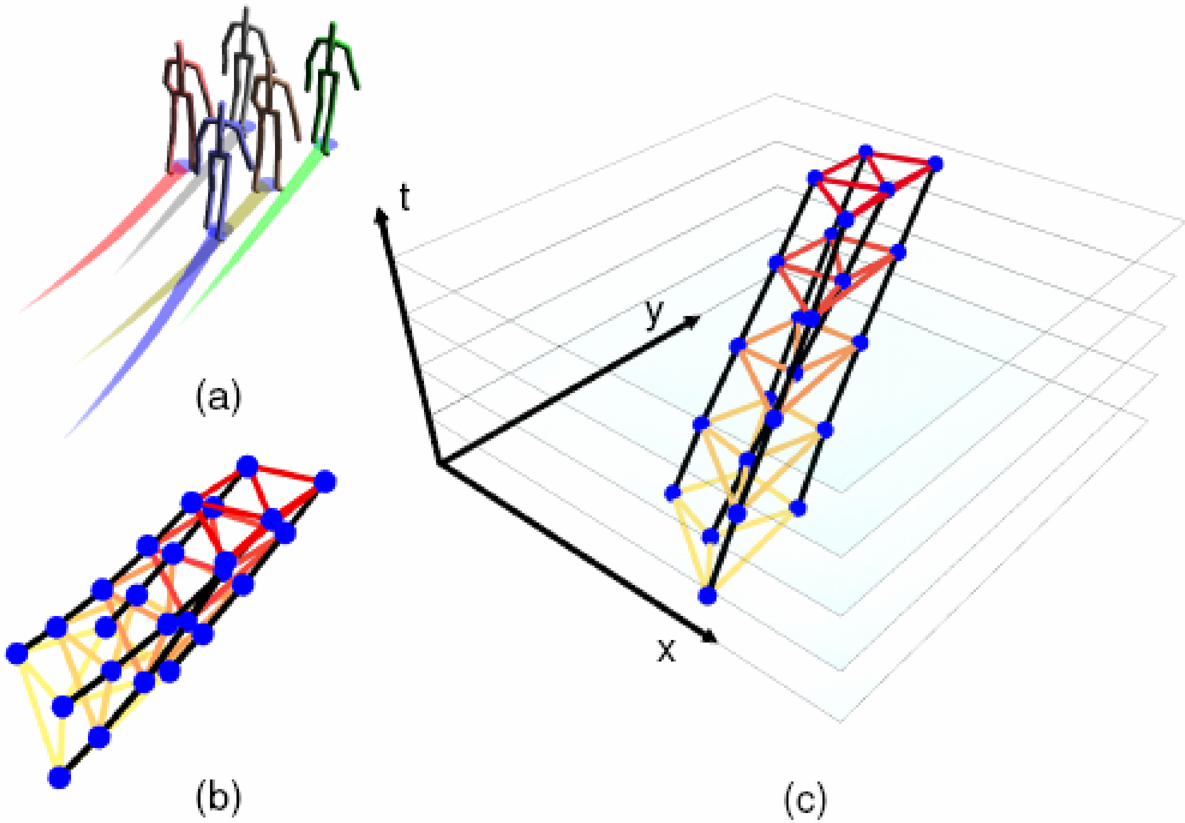


Figure 2: *Graph representation of a group motion clip. (a) A short group motion clip. (b) A graph constructed from the clip. (c) Conceptual view of the graph. The graph encodes time-varying group formations.*

Local feature (1)

- Local feature c for a vertex v is defined as an ordered triple of vertices (u, v, w), where u and w are adjacent to v.

$$\mathbf{v} = \mathbf{u} + c_x(\mathbf{w} - \mathbf{u}) + c_y\mathbf{R}(\mathbf{w} - \mathbf{u}) \quad (1)$$

- For notational convenience, define a function f_c that returns the desired location of v by linearly combining adjacent vertices u and w based on the local

$$\mathbf{f_c}(\mathbf{u}, \mathbf{w}) = \mathbf{u} + c_x(\mathbf{w} - \mathbf{u}) + c_y\mathbf{R}(\mathbf{w} - \mathbf{u}). \quad (2)$$

Local feature (2)

- To measure the difference between any original feature c = (u, v,w) and its deformed feature c' = (u', v',w'), the distortion metric D(·) is defined by using the squared distance between f_c(u', w') and v as follows:

$$D(\mathbf{c}, \mathbf{c}') = \|\mathbf{f}_{\mathbf{c}}(\mathbf{u}', \mathbf{w}') - \mathbf{v}'\|^2. \qquad (3)$$

Issue with D(.)

- Unnaturally enlarged, or shrunken the deformation of the original feature.
- Igarashi et. al's two step optimization scheme in which two least-squares problems are solved sequentially.
- IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2005. As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics (SIGGRAPH 2005) 24, 3, 1134–1141.*

Two-step algorithm

- Laplacian deformation: an input graph G is first deformed to yield an intermediate graph G’ that has minimal overall distortion of local features while allowing free translation, rotation, and uniform scaling (see Figure 3(a) and (b)).
- Unnecessary scaling effects are then compensated to produce the final deformed graph G’’ in the subsequent scale-adjustment step (see Figure 3(c)).

Two-step algorithm

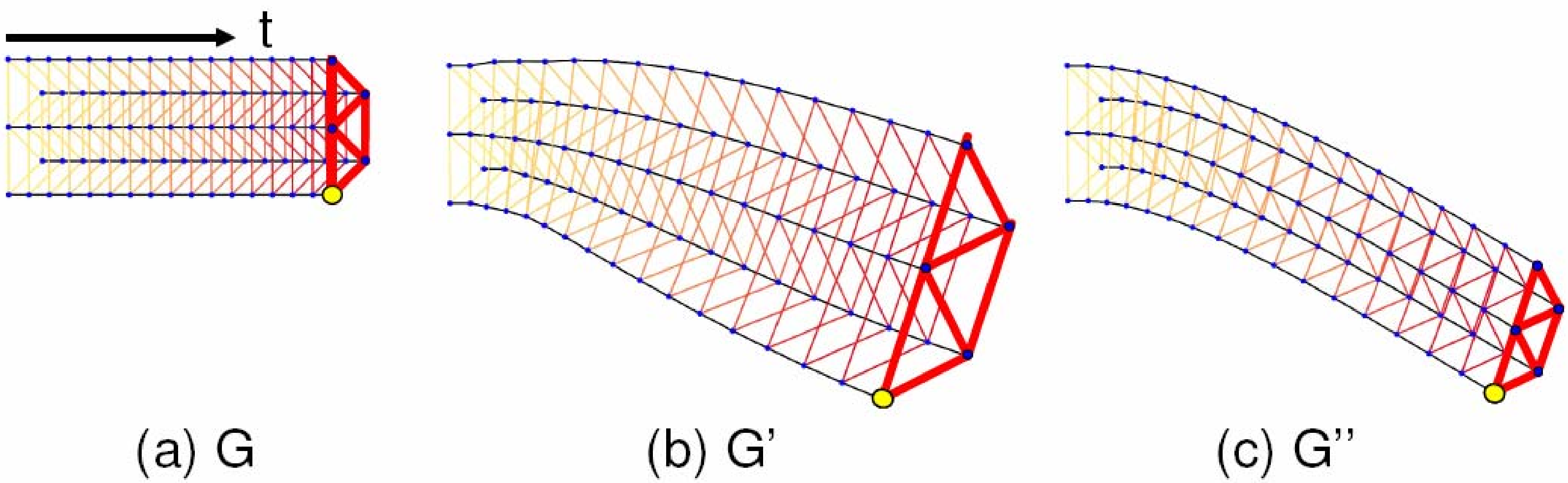


Figure 3: *Graph deformation. In source animation (a), five individuals are marching in two rows. The source animation is deformed by dragging the yellow-colored vertex. (b) Laplacian deformation could result in unnatural scaling artifacts. (c) The scale compensation step alleviates the artifacts.*

Step 1: Laplacian deformation (1)

- A <u>temporal feature</u> is defined by a pair of adjacent motion edges.
- A <u>spatial feature</u> of vertex v is defined by an adjacent triangle lying on the time plane containing v.
- A <u>spatiotemporal feature</u> is defined by a motion edge and a formation edge.

Step 1: Laplacian deformation (2)

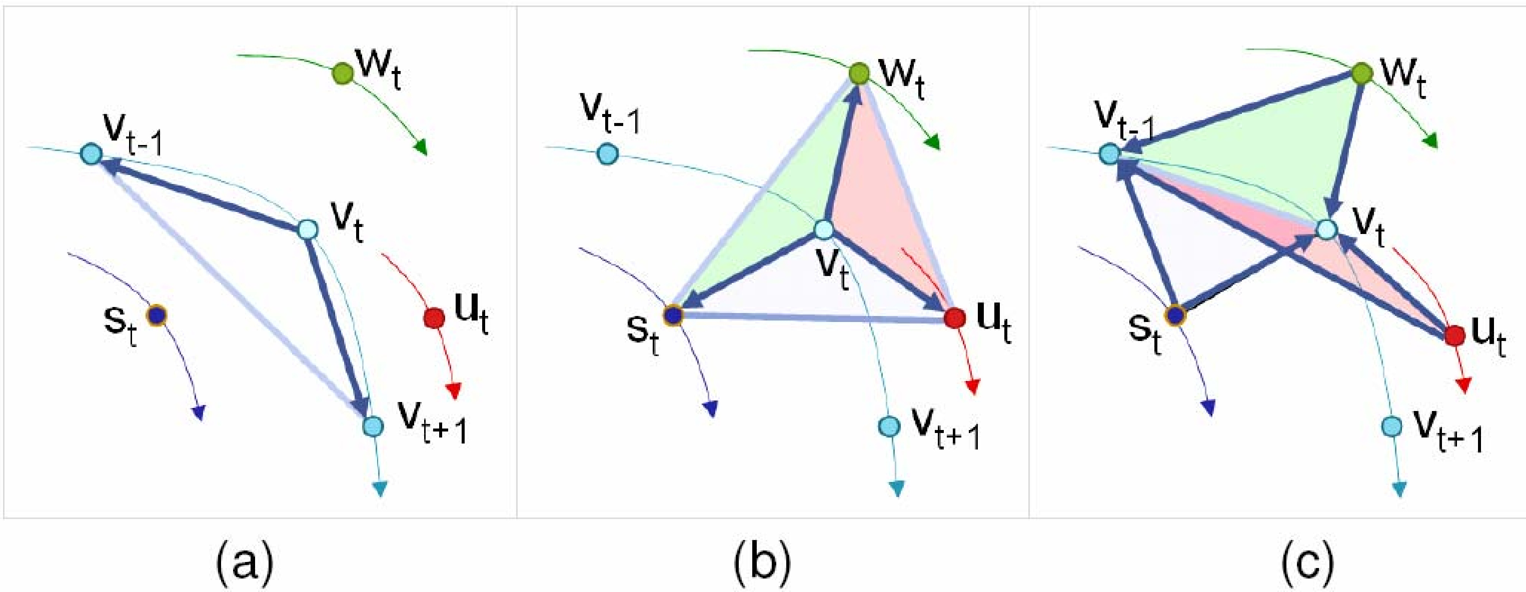


Figure 4: *Three types of triangular features. (a) Temporal feature. (b) Spatial feature. (c) Spatiotemporal feature.*

Step 1: Laplacian deformation (3) Objective function

$$E_1(\mathbf{G}') = E_S(\mathbf{G}') + \alpha E_T(\mathbf{G}') + \beta E_{ST}(\mathbf{G}') \qquad (4)$$

where $E_S(\cdot)$, $E_T(\cdot)$, and $E_{ST}(\cdot)$ measure the distortions of spatial, temporal, and spatiotemporal features, respectively. The weight values alpha and beta control the relative importance among them.

Step 1: Laplacian deformation (4) Objective function

- $E_S(\cdot)$ measures the distortion of neighborhood formations.

$$E_S(\mathbf{G}') = \sum_{\mathbf{c}' \in \mathbf{C}'_S} w(\mathbf{c}) \|\mathbf{f}_{\mathbf{c}}(\mathbf{u}', \mathbf{w}') - \mathbf{v}'\|^2, \qquad (5)$$

- C'_S is a set of spatial features in which both u' and w' are connected to v' by formation edges, and w(c) is the inverse of $\|\overrightarrow{\mathbf{vu}}\|^2 + \|\overrightarrow{\mathbf{vw}}\|^2$.

Step 1: Laplacian deformation (5)
Objective function

- E_T (·) reflects the distortion in individual moving trajectories. C'_T is a set of temporal features in which both u' and w' are connected to v' by motion edges.

$$E_T(\mathbf{G}') = \sum_{\mathbf{c}' \in \mathbf{C}'_T} \|\mathbf{f_c}(\mathbf{u}', \mathbf{w}') - \mathbf{v}'\|^2, \qquad (6)$$

Step 1: Laplacian deformation (6)
Objective function

- $E_{ST}(.)$ represent the spatiotemporal relationships among vertices. C'_{ST} is a set of spatiotemporal features in which v' is connected to u' and w' by a formation edge and a motion edge, respectively.

$$E_{ST}(\mathbf{G}') = \sum_{\mathbf{c}' \in \mathbf{C}'_{ST}} \|\mathbf{f}_{\mathbf{c}}(\mathbf{u}', \mathbf{w}') - \mathbf{v}'\|^2, \qquad (7)$$

Step 2: Scale compensation (1)

- Fit every spatial feature in G to best match the corresponding spatial features in the intermediate graph G'. The least-square fitting of each feature can be achieved by minimizing (see [Igarashi et al. 2005] for details):

$$f_{\mathbf{c}} = \|\mathbf{v}^F - \mathbf{v}'\|^2 + \|\mathbf{u}^F - \mathbf{u}'\|^2 + \|\mathbf{w}^F - \mathbf{w}'\|^2, \qquad (8)$$

where c is a spatial feature and vF is constrained to satisfy vF = fc(uF ,wF).

Step 2: Scale compensation (2)

- The deformed graph can be obtained by minimizing

$$E_2(\mathbf{G}'') = E_{SC}(\mathbf{G}'') + \alpha' E_T(\mathbf{G}'') + \beta' E_{ST}(\mathbf{G}''). \qquad (9)$$

- Where

$$E_{SC}(\mathbf{G}'') = \sum_{\mathbf{c}'' \in \mathbf{C}''_F} \left(\|\overrightarrow{\mathbf{v}''\mathbf{u}''} - \overrightarrow{\mathbf{v}^F\mathbf{u}^F}\|^2 + \|\overrightarrow{\mathbf{v}''\mathbf{w}''} - \overrightarrow{\mathbf{v}^F\mathbf{w}^F}\|^2 \right).$$

Two-step algorithm (summary)

- Given the objective functions E_1 and E_2, the graph deformation process is formulated as a two-step constrained least-squares optimization problem.
- employ a Lagrange multiplier scheme to solve a sparse linear system.

Stitching Group Motions (1)

- Given two graphs G and G' having the same number of individuals N, we build a new graph G'' that concatenates two group motion clips sequentially in time. G'' has $T = t + t' - 1$ time frames, where t and t' are the time frames of G and G', respectively.
- Stitching two graphs requires three steps.

Stitching Group Motions (2)

- Three steps:
  - Establish one-to-one correspondences between individuals in two groups (see Figure 6(a)).
  - Two motion clips should be aligned via rigid transformation (see Figure 6(b)).
  - two group motions are concatenated by smoothly morphing group formations at the boundary (see Figure 6(c)).

Stitching Group Motions (3)

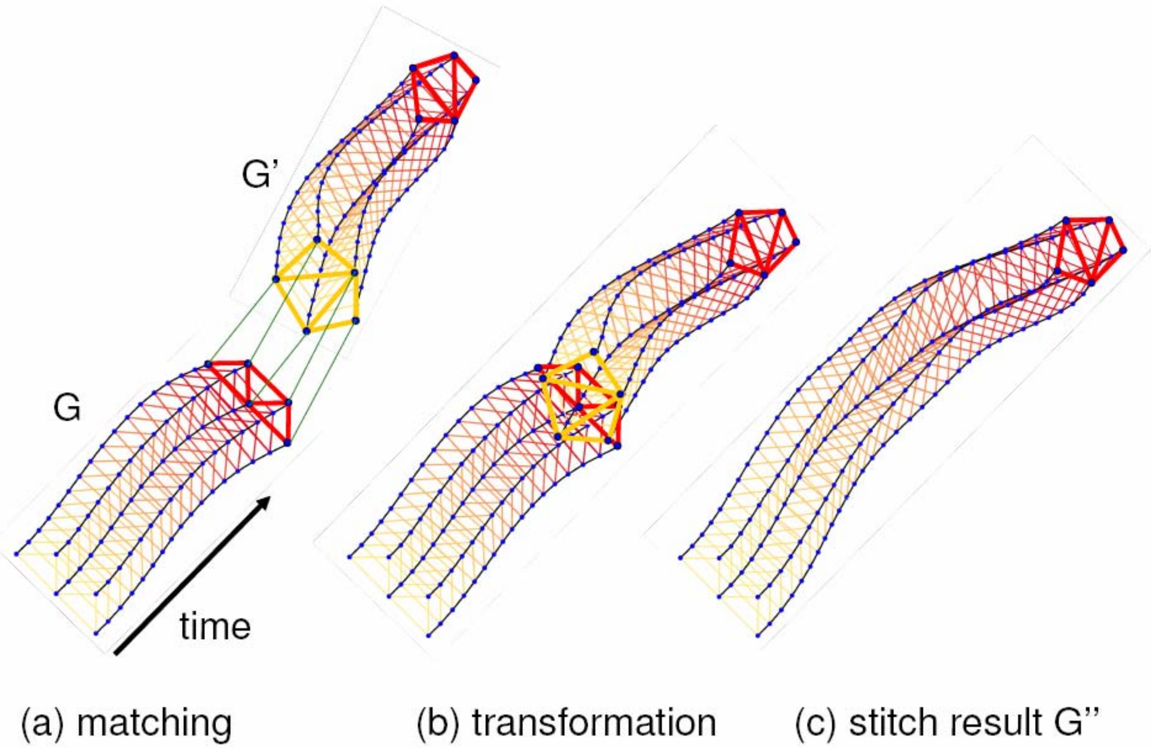


Figure 6: *Group motion stitching.*

Stitching Group Motions (4)

- Step 1 is formulated as a bipartite graph matching algorithm in BELONGIE, S., MALIK, J., AND PUZICHA, J. 2002. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell. 24, 4, 509–522.*
- Step 2 is to align two motion clips by translating and rotating them to best match the boundary by the approach in KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. *ACM Transactions on Graphics (SIGGRAPH 2002) 21,* 3, 473–482.

Stitching Group Motions (5)

- The final step of stitching is to concatenate two group motions by smooth blending of group formations.
- Linear blending individual trajectories cannot give desired result.
- Instead, blend three triangular features, which generates a smooth transition of the group formation while maintaining the detail characteristics of individual moving trajectories.

Stitching Group Motions (6) step 3: blending three features (1)

- Blending temporal features
  - Copy the temporal features of G to the first ($t-1$) frames of concatenate motion G'' and temporal feature of G' to the last ($t' - 1$) frames of G''.
  - Temporal features at intervening frame t are obtained by linearly interpolating the temporal features at frame $t-1$ and frame $t + 1$.

Stitching Group Motions (6) step 3: blending three features (2)

- Blending spatial and spatiotemporal features
  - compute an intermediate graph $\bar{\mathbf{G}}$.

    Let $\{\mathbf{w}_{i,j}\}_{j=1}^{N}$ be the frames of $\bar{\mathbf{G}}$.

    For frame i < t,

    $$\mathbf{w}_{i,j} = \mathbf{v}_{i,j} + f_a(t-i)(\mathbf{w}_{t,j} - \mathbf{v}_{t,j}),$$

    $$f_a(\Delta t) = 1 - 3\left(\frac{\Delta t}{t}\right)^2 + 2\left(\frac{\Delta t}{t}\right)^3. \qquad (11)$$

  - copy the spatial and spatiotemporal feature of $\bar{\mathbf{G}}$ to G''.

Postprocess (1)

- Collision avoidance - the deformation or stitching of a group motion can lead to collision or insufficient clearance between individuals.
- Time warping to handle irregular speed change.

Postprocess(2) -Collision avoidance

– Approximating each individual trajectory by a time-parameterized piecewise linear curve;
– If two trajectories are closer than a certain threshold at any time instance, pulls the trajectories away by 10% of the threshold and repeats this process until all collisions are

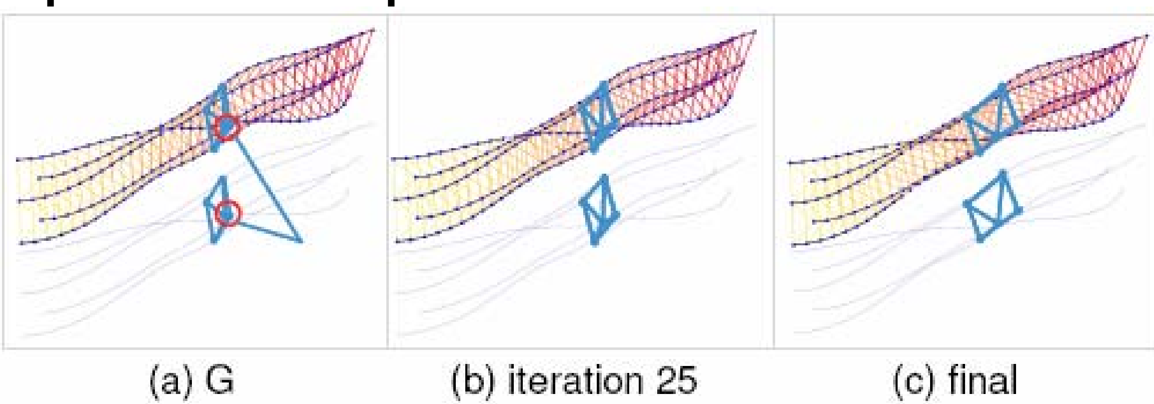


Figure 7: *Collision avoidance for a small ($N = 5$) group of characters.*

Postprocess(3) -Collision avoidance

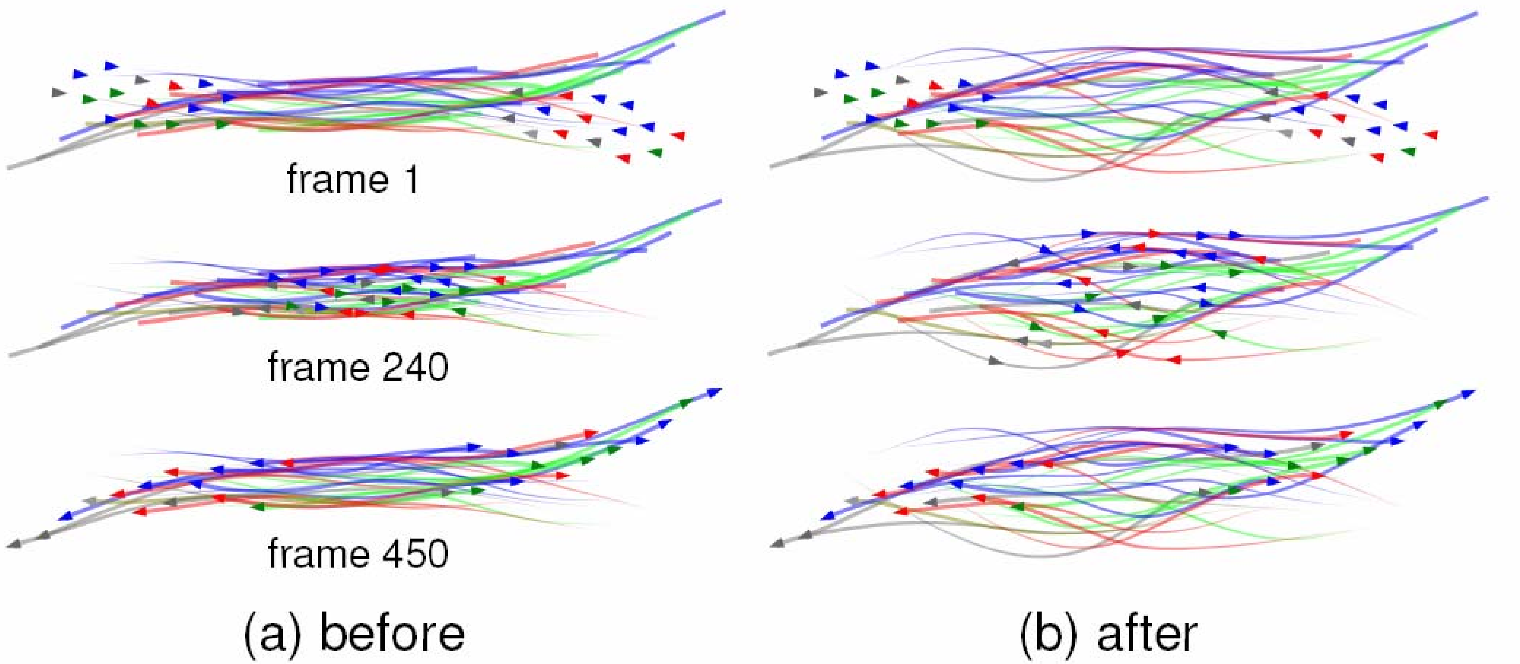


Figure 8: *Collision handling. (a) Two groups of crowds are bumping into each other. (b) Collisions are resolved iteratively while fixing the original formations at the first and last frames. Resolving all collisions required 1997 iterations, which took about four seconds.*

Postprocess(4)
-Collision avoidance

with obstacles

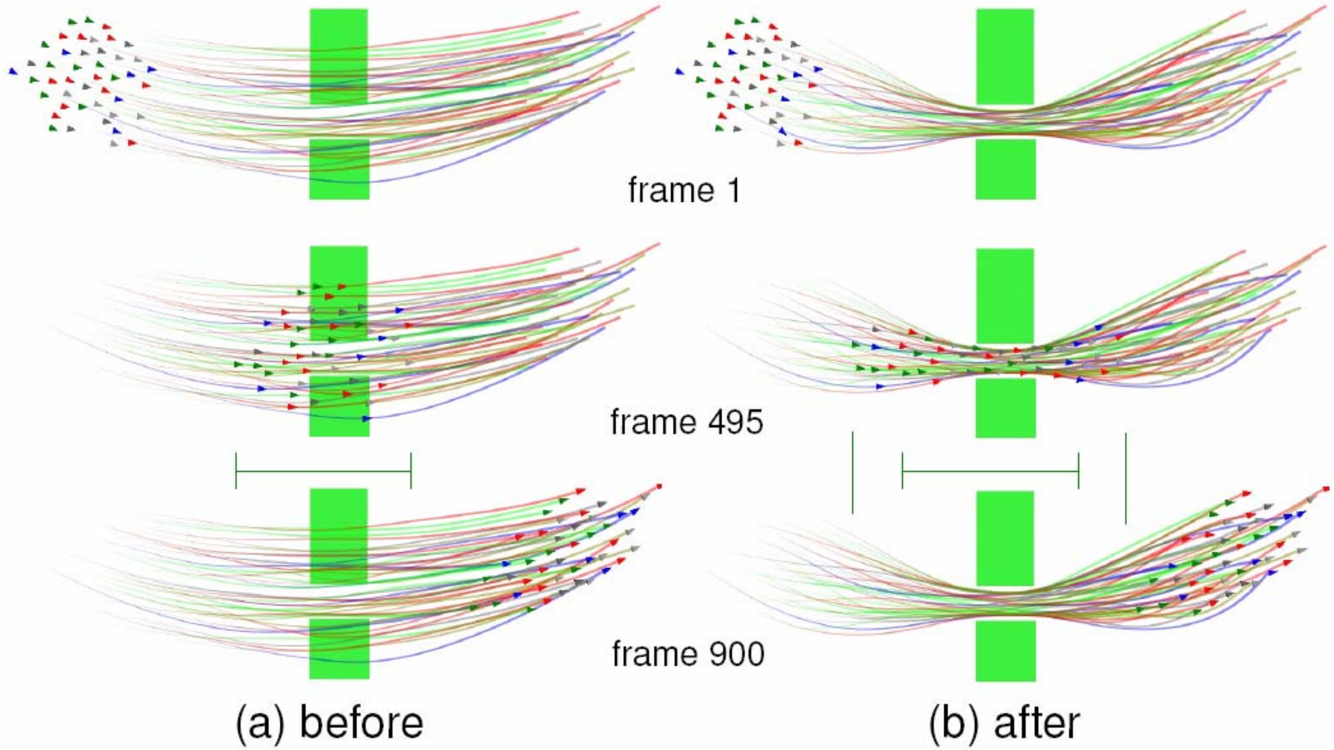


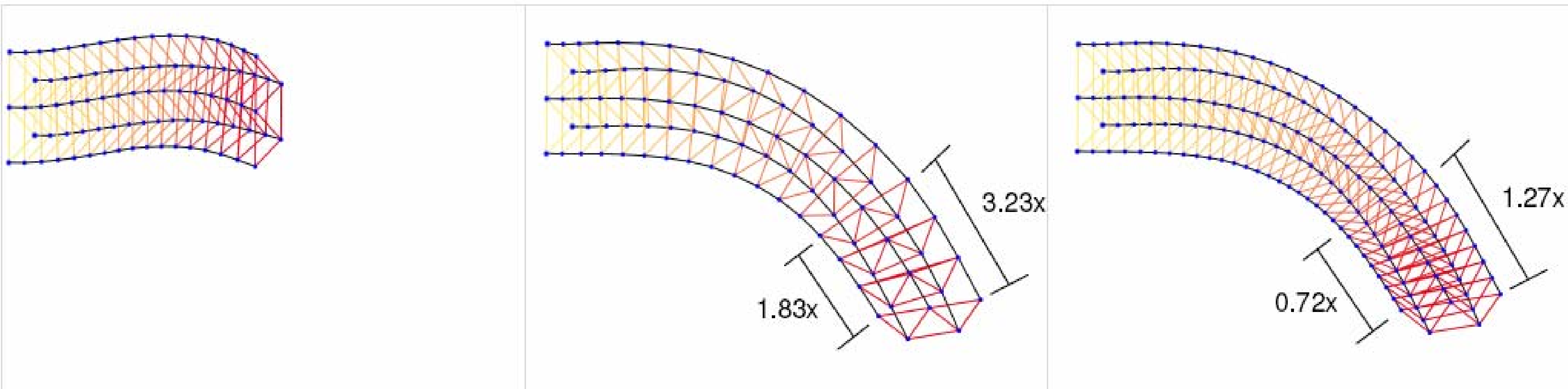
Figure 9: *Forcing a circular group (a) through a narrow opening such that the group must elongate to pass. In this specific example, penetrating points inside obstacles are pulled toward the opening. Resolving all collisions required 34733 iterations, which took about 163 seconds.*

Postprocess (5)- Time warping

- Undesired speed change.
- Goal - allow characters in a deformed group motion clip to move as closely to their original speeds as possible.
- Model the issue as a least-squares optimization.

$$E_t = \sum_{i=1}^{T} \sum_{j=1}^{N} \left(\frac{\|\mathbf{v}'_{i+1,j} - \mathbf{v}'_{i,j}\|}{\Delta t'_i} - \frac{\|\mathbf{v}_{i+1,j} - \mathbf{v}_{i,j}\|}{\Delta t} \right)^2, \quad (12)$$

Postprocess (6)- Time warping



(a) G (b) G'' (c) after timewarping

Figure 10: *Timewarping. (a) Source animation. (b) Due to deformation, the individual trajectories are stretched and thus the characters on the deformed trajectories have to walk faster than on the original trajectories. (c) Timewarping re-parameterizes the trajectories to minimize excessive speedup and slowdown.*

Results

- Two-group bumping
- Battlefield
- Town

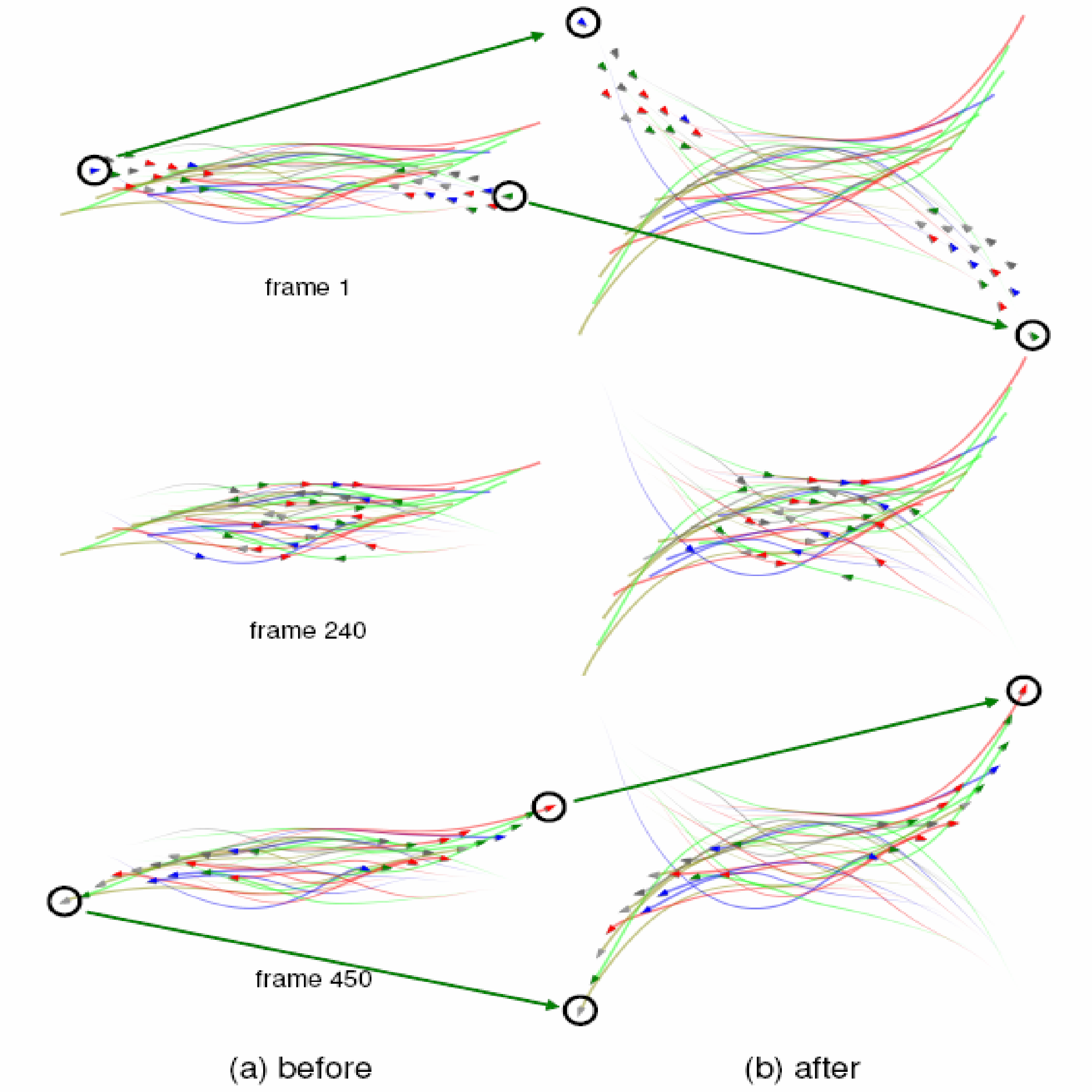


Figure 11: *A group of characters moving in two different directions. (a) Original motion. (b) Editing by dragging four vertices.*

Performance

Table 1: *Statistics on the final group motion clips. The maximum computation time was measured by performing a deformation operation on the largest group in the scene.*

	# of frames	# of sample frames	# of characters	# of groups	total # of vertices	# of vertices in the largest graph	maximum computation time
Two-groups bumping	450	15	30	1	450	450	219ms
Battlefield	3958	132	50	2	6600	6600	3803ms
Downtown	900	30	348	15	10440	1500	677ms

Limitations

- A large deformation of a group motion can lead to unnatural speedup/slowdown of individual motions.
- The time-warping scheme mitigates such artifacts to some extent, there are cases where characters cannot maintain their original formation unless some characters move extremely fast. The authors allow the user to handle such cases interactively.