Computational Aesthetics in Graphics, Visualization, and Imaging (2012), pp. 1–8
D. Cunningham and D. House (Editors)

SAMBA: Steadied choreographies

Jarek Rossignac[1], Mark Luffel[1] and Àlvar Vinacua[2]

[1]School of Interactive Computing, College of Computing, Georgia Institute of Technology
[2]Technical University of Catalonia, Spain

Abstract

Given the start positions of a group of dancers, a choreographer specifies their end positions and says: "Run!" Each dancer has the choice of his/her motion. These choices influence the perceived beauty (or grace) of the overall choreography. We report experiments with an automatic approach, SAMBA, that computes a pleasing choreography. Rossignac and Vinacua focused on affine motions, which, in the plane, correspond to choreographies for three independent dancers. They proposed the inverse of the Average Relative Acceleration (ARA) as a measure of grace and their Steady Affine Morph (SAM) as the most graceful motion. Here, we extend their approach to larger groups. We start with a discretized (uniformly time-sampled) choreography, where each dancers moves with constant speed. Each SAMBA iteration steadies the choreography by tweaking the positions of dancers at all intermediate frames towards corresponding predicted positions. The prediction for the position of dancer at a given frame is computed by using a novel combination of a distance weighted, least-squares registration between a previous and a subsequent frame and of a modified SAM interpolation. SAMBA is fully automatic, converges in a fraction of a second, and produces pleasing and interesting motions.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [CHECKALL Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Our long-term goal is to devise: (1) a mathematical formulation that measures the beauty (or grace) of a choreography and (2) a practical algorithm for computing the most beautiful choreography, given a set of constraints or choreographer's directives. In the present paper, we focus on a specific sub-problem: iteratively improving the beauty of a planar motion of a small group of particles, given a time interval and their initial and final positions. These particles may represent the instantaneous positions of a small group of dancers, hence, we will use a terminology derived from this metaphor, even though we have not validated the benefit of our approach for this application domain.

We consider a group of n dancers, each represented by a point on the plane. Let ${}^tP_i$ denote the position of dancer i at time t. Without loss of generality, we assume that t varies between 0 and 1. We are given the start position ${}^0P_i$ and the end position ${}^1P_i$ of each dancer i. We wish to compute their motions. The term choreography will refer to the combined set of these motions.

Our algorithm discretizes the choreography, representing it by a set of frames at evenly spaced time samples. Each frame is associated with a time t and defines the instantaneous positions ${}^tP_i$ of each dancer. Continuous motions may be obtained by computing an interpolating spline or subdivision curve for each dancer.

The perception of beauty of a choreography is clearly subjective and often influenced by expertise (being a dancer versus a computer...programmer), priming (watching modern dance versus soccer), and context (accompanying music). Still, objective guidelines for designing beautiful motions have been offered by members of the artistic community.

The first guideline of interest states that straight-line, constant-velocity motions should be avoided [TJ95, WNS*10]. This may be seen as an extension of early guidelines proposed by Hogarth, who stated that "straight lines ... signify stasis, death, or inanimate objects" [Hog53]. The second guideline, voiced by Chekhov in a letter to Maxim Gorky claims that grace is inversely proportional to the amount of superfluous movement: "When a person expends the least amount of motion on one action, that is grace" [Che76].

Combining these two guidelines, we seek a choreography that is the simplest (in some sense), and yet not made of linear motions. When the group has a single dancer, these two guidelines seem incompatible and a linear interpolation appears to be the only natural choice, as seen in fig. 1a). For a pair of dancers at constant motion a possible choice would be a circular motion (fig. 1b).

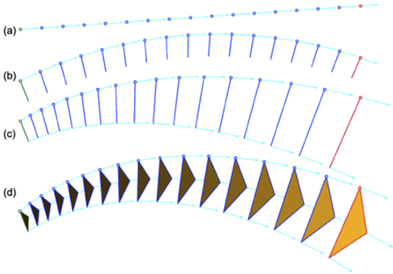

Figure 1: *For one dancer, a linear motion (a) is the natural choice. For a group of two dancers at constant distance of each other, we advocate a pure rotation (b). When the distance between the dancers is different at the start and end frames, we advocate a logarithmic spiral (c). For 3 dancers, we advocate a SAM (d).*

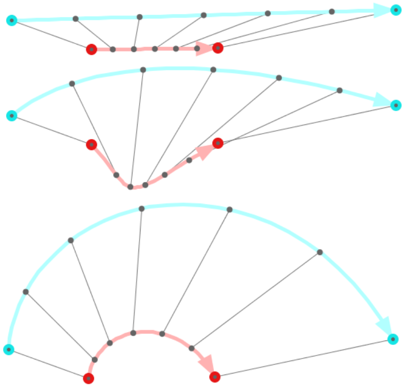

Figure 2: *Comparison 1 of the three motions.* Linear *(top),* Dynamic *(middle),* Spiral *(bottom).*

To gain some insight and appreciation of the problem at hand, let us focus on a group of two dancers and compare three choreographies that interpolate the same set of constraints: Linear choreography (fig. 2 top), Dynamic (fig. 2 middle), and Spiral (fig. 2 bottom). These are also compared in the accompanying video.

Linear minimizes travel distance and moves each dancer with constant velocity:

$$ {}^{t}P_i = (1-t)\,{}^{0}P_i + t\,{}^{1}P_i \qquad (1) $$

Unfortunately, this solution is often unacceptable because animators abhor straight-line and uniform-speed motions (as mentioned above), but it serves as a starting point of the SAMBA steadying process. Hence we include it in our comparison.

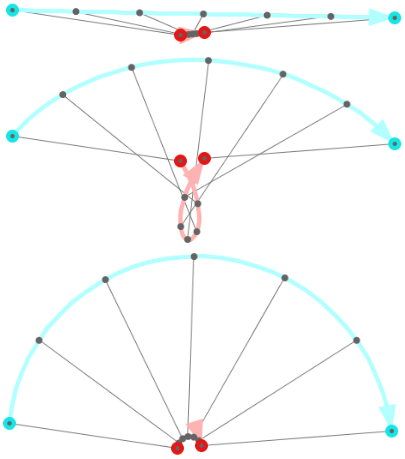

Figure 3: *Comparison 2 of the three motions.* Linear *(top),* Dynamic *(middle),* Spiral *(bottom).*

Dynamic assumes an interpolating motion that preserves linear and angular momenta. In our metaphor, this choreography simulates a two-body motion free from external forces and torques, but where the two dancers pull or push on each other to change the distance that separates them. For simplicity, we have chosen to vary that distance linearly, but other options could be explored. We include it in this comparison, because physical plausibility is a natural option for defining optimal choreographies. We discuss below some of its aesthetic drawbacks.

Spiral computes a fixed point of the similarity transform that maps the start frame to the end frame (fig. 1c) and uses a synchronized constant angular speed rotation with an exponential scaling, both about the fixed point. We provide details further in the paper. We include this option in this preliminary discussion, because it captures the spirit and some of the advantages of the proposed solution.

These three choreographies look very different from each other (especially on video). Let us point out more formally the similarities and differences between these three choreographies.

The distance between the dancers varies linearly in Dynamic, exponentially in Spiral, and possibly non-monotonically in Linear. The latter is probably the most important drawback of Linear.

The midpoint (center of mass) of the dancers moves with constant velocity in Linear and Dynamic. This linear motion of the center of mass violates the artistic principle discussed above. Spiral moves the center of mass along a more pleasing arc.

LINEAR is often unacceptable, because the straight line motion of each dancer appear too mechanical and because it portrays a rather selfish choreography, where each dancer appears uninterested in the behavior of other dancers.

DYNAMIC may be the proper choice when the dancers hold each other and are swirling on the dance floor or on ice (without skates). But the overall motion of (the center of mass of) the couple is uninterestingly linear and the choreography seems passive, lacking determination and energy.

Assume that the line segment drawn between the two dancers represents their arms (as if they were holding hands) or the direction of their gaze towards each other. Notice that the angle between this line segment and the instantaneous velocity of each dancer varies in both LINEAR and DYNAMIC, but remains constant in SPIRAL. Also note that the orientation of the line segment between the dancers rotates at constant speed in SPIRAL, slows down as distance increases in DYNAMIC, and can be more chaotic (non-monotonic) in LINEAR. For these reasons, we believe that SPIRAL is perceived as being more harmonious than the other two.

The advantage of SPIRAL over the other two approaches becomes even more obvious if one translates the two end positions so that the start and end positions of one dancer (say A) coincide. In that case (as shown in fig. 3 and accompanying video), LINEAR leaves A in place, while moving the other dancer (say B) along a straight line, which may pass close to A. DYNAMIC moves both A and B. In fact, the motion of B seems unnecessarily complex. SPIRAL leaves A in place and moves B along a spiral at a constant angular speed around A. In fact, if the distance between the two dancers is the same in the start and end frames, it remains constant throughout the choreography and B moves along a circular arc (1 b). We conjecture that this behavior would be viewed by most observers as the most natural of the three.

In the remainder, we extend this SPIRAL behavior to larger groups of dancers.

We claim three novel contributions:

1. We propose to measure the grace of the choreography by the integral over time (approximated by the normalized sum over all intermediate frames and dancers) of a local measure of steadiness. This new measure is an extension (to non-affine motions) of the steadiness measure originally proposed in [RV11], which was restricted to affine motions (fig. 1d). We provide a mathematical definition and an efficient algorithm for this extended measure of steadiness.
2. We propose a novel formulation of the local, instantaneous approximation of the choreography around any given dancer at an intermediate frame. We call it the Local Instantaneous Steady Affine Motion Approximation (LISAMA) and define is using a Steady Affine Morph (SAM) or a Steady Logarithmic Spiral Morph between a preceding and a succeeding frame. LISAMA extends the classical model of an instantaneous velocity approximation of the motion of a single point to the instantaneous affine motion.
3. We propose a novel approach, called SAMBA, for computing an initial choreography and for improving its steadiness through a series of steadying steps. A steadying step first estimates the position of each dancer in each intermediate frame using LISAMA. Then it moves these positions (half-way by default) towards their estimates. Successive passes increase the steadiness of the choreography and may be executed with increasing (temporal) locality to accelerate convergence. We say that SAMBA produces a steadied choreography that morphs between two frames of a group of dancers.

Several stages of a typical SAMBA process and the resulting steadied choreography are shown in figure 4.

In the remainder of the paper: (1) we define our notation more precisely and summarize prior work on steady motions, (2) we review relevant prior art, (3) we provide the details of LISAMA, (4) we present the SAMBA algorithm, and (5) we show results and discuss limitations.

2. Notation and Background

Let ${}^{t}P_i$ denote the position at time t of dancer i, where i is an integer between 0 and $n-1$. We are given the initial position ${}^{0}P_i$ and the final position ${}^{1}P_i$ of each dancer P_i. (Note that we use a preceding superscript to identify a time or frame.) We wish to compute its motion ${}^{t}P_i$, which maps time t to position ${}^{t}P_i$.

For simplicity and without loss of generality, we have reparameterized time so that the desired morph starts at time 0 and ends at time 1.

Here, we are interested in choreographies that exhibit a certain degree of continuity in the time evolution of a local arrangement of nearby dancers. Hence, we propose to measure the quality of a choreography by its steadiness. The concept of steadiness has been defined for affine motions [RV11]. We extend it here to non-affine choreographies.

The motion of a group of points is affine if, there exists a continuous function that maps time t to an affinity ${}^{t}A$ such that ${}^{t}P_i = {}^{t}A{}^{t}P_0$ for all points.

The motion is steady if an affinity A exists such that

$$ {}^{t}A = A^{t} \tag{2} $$

Here t is a real number between 0 and 1 and the notation A^t defines the non-integer power (also called root) of an affinity. Closed-form solutions in two and three dimensions that compute a real matrix A^t, when it exists, from the matrix representation of affinity A have been proposed in [RV11]. In the rare situations where a real matrix A^t does not exist, one

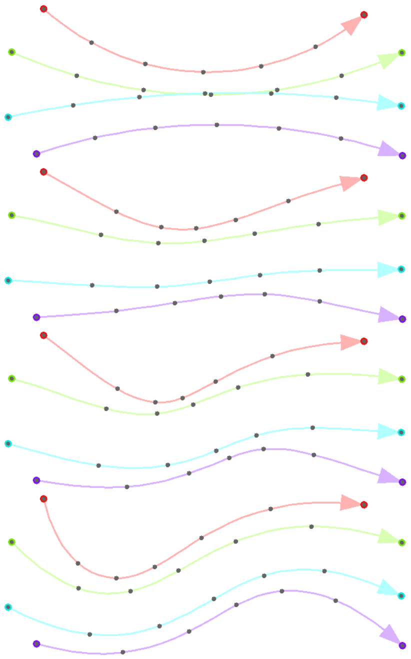

Figure 4: *Given the initial (left) and final (right) key frames defining the start and end positions of a group of dancers in the plane, SAMBA computes the intermediate frames of an interpolating morph using an iterative process that strives to increase steadiness. We show dancers' positions at evenly time-spaced frames (black dots along each trajectory). We show the result (from top to bottom) after one SAMBA pass, after 100, 500, and 1500 passes (bottom). We used 30 intermediate frames during this computation (even though only 5 dots are shown per trajectory), The whole process took 23 milliseconds to complete. We also show in light grey line segments between selected pairs of dancers, as if each dancer were holding hands with immediate neighbors. Notice how increasing steadiness increases the smoothness and regularity of these grey curves.*

must use an unsteady alternative. In this cases we default to a SAM between two affinities, which are related by a similarity transform.

Given the start and end positions of a group of 3 dancers, the Steady Affine Morph (SAM) proposed by Rossignac and Vinacua [RV11] computes an interpolating steady motion (fig. 1d).

To compute the position of the three dancers at time t, SAM computes the affinity B that brings the points (0,0),(1,0),(0,1) to their corresponding positions in the start frame and the affinity C that brings the same points to their corresponding positions in the end frame. Then it computes $A = CB^{-1}$ and finally ${}^{t}\!A$ as $A^{t}B$.

An example of a SAM choreography is shown in figure 1, which also illustrates that SPIRAL, pure rotation, and pure translations are special cases of SAMs.

A measure of steadiness was introduced by Rossignac and Vinacua [RV11] as the inverse of the Average Relative Acceleration (ARA). They define ARA as the integral over space and time of the acceleration with respect to a local frame. In the present context, lacking such a local frame, we measure ARA as the 2-norm of the difference ${}^{t}\!A - A^{t}$. Hence, for a steady motion, ARA $= 0$.

The difficulty of using ARA to define steadiness of a choreography of dancers in a larger group is that the choreography of a group is in general not affine. Hence, our solution uses a LISAMA to compute a predictor ${}^{t}Q_i$ of the location of each dancer ${}^{t}P_i$ in an intermediate frame from the positions of this same dancer and its neighbors in the previous and next frames, and returns (as a measure of steadiness) the sum of the squared deviations $({}^{t}Q_i - {}^{t}P_i)^2$.

3. Prior Art

Several computer vision techniques developed for video compression or segmentation and for shape or camera tracking [MV98, Mod04, GKB*07] compute affine displacement maps (registration) [KC96, KC98, Kru98] that maximize color or gradient correspondence between one portion of the image in one frame and a portion in a subsequent frame. As such, they build a local affine approximation [BA96] or hierarchies of these [ACM*10] of the choreography between two frames. The problem addressed in this paper is somewhat orthogonal to these computer vision problems: In our case correspondence is given (as pairs of initial and final positions of particles) and the goal is not to produce a local affine map (that part is standard [BLCD02, SMW06]), but to compute the detail of the interpolating choreography, i.e., the intermediate (non affine) trajectories for all the particles. Our solution may be of interest and potentially applicable to computer vision for producing smooth slow motions, i.e., generating inter-frame images.

Several morphing techniques have been studied, where the particles are samples of a lower-dimensional manifold, i.e., vertices of a curve in 2D or of a surface in 3D. Some of these techniques establish correspondence from the orientation of the manifold [KR92], from proximity [ESE06], or from both [CLRW10]. Other techniques assume a given correspondence [SG92, SGWM93]. Some morphing techniques use linear trajectories. The ball morph [WR09, WR10] generates circular trajectories that are orthogonal to the initial and final manifolds. BetweenIT [WNS*10] generates logarithmic spirals or blends of such spirals. Some approaches are focused on the evolution of the shape and of its orientation, but not on the actual motion followed by the shape during the morph [SGWM93, Kor02]. We cannot directly benefit from these solutions, because our group of particles are not samples of a manifold and hence do not remain aligned along a curve or surface throughout the motion.

Several morphing techniques establish a triangular or rectangular lattice that either connect the particles [ACOL00] or surrounds them [MHTG05, RJ07, SDC09]. They endow these full dimensional cells (triangles or quads) with stiffness or other rigidity properties and evolve them from their initial configuration to their final configuration while striving to keep them rigid or to make them each evolve in a continuous manner. These techniques are most effective when the initial and final configuration have a common triangulation [SG01] or quadrangulation, and when the desired deformation is free from self-overlaps. We are interested in supporting more general choreographies, where this may not be the case.

Particle based fluid simulation techniques [MSKG05] advect the particles striving to simulate an incompressible viscous flow. For that, they compute accelerations from local measures of particle density and friction. The flow is the result of a forward dynamic simulation. Hence, these techniques cannot be used directly to solve our morphing problem, although we plan to explore using SAMBA for artistic design of fluid flows.

Tools for designing animations of crowds [JCP*10, THL*04], flocks [Rey87], or schools [TT98] may benefit from SAMBA in situations where the initial and final positions of all of the particles, or at least of some "leader" particles, are either provided by the artist or captured from sensors or cameras. A particularly effective solution for designing such animations was proposed by Kwon et al. [KLLT08]. They build a (possibly different) Delaunay triangulation of the particles at each frame, but allow the artist to edit the resulting "formation edges" interactively. They connect each particle of an intermediate frame to the corresponding particle in the previous and next frame, hence linking successive per-frame triangulations through these "motion edges". Then they let the user deform this multi-frame graph by moving and pinning any vertex of that graph. They use the As-Rigid-As-Possible shape manipulation technique [IMH05] to minimize the distortion from the original graph while satisfying the "pinned" constraints. Their approach minimizes a distortion metric that sums the squares of the differences between the current and initial locations of each particle, expressing these locations in a local frame defined by three neighboring particles. They solve for the new configuration using a constrained least square optimization. They also propose further improvements (scale-free Laplacian and post-processing scale-compensation) to reduce undesirable effects of local scaling and distortions near degenerate triangles. Their approach is particularly effective when it is desired to maintain the relative (local) formations of nearby particles throughout the choreography. Our SAMBA solution does not rely on instantaneous triangulations, and hence does not suffer from artifacts that may occur when the connectivity (formation edges) changes from one frame to the next. Furthermore, it does not require the choreographer to specify the original graph. (In our case, the trajectories are computed automatically from the initial and final locations.)

The continuum crowds method [TCP06] produces crowd motion by tracing particles through an evolving potential field generated from user-defined goals, scalar cost field, and the positions and velocities of other agents. The potential field is stored on a grid and computed by the fast marching method per group of agents that share a common goal.

The SAMBA solution proposed here builds upon ideas from these different fields, but differs significantly from them. In particular, it does not assume or compute a connectivity graph.

4. LISAMA

Our Local Instantaneous Steady Affine Motion Approximation (LISAMA) computes a steady motion that approximates the choreography near the position ${}^{\beta}P_i$ of a given dancer P_i at time β of an intermediate frame. In what follows we use $a, b, c, \ldots$ to refer to frames (numbered from 0 to n), and $\alpha, \beta, \gamma, \ldots$ for their respective times.

First, we compute weights ${}^{\beta}w_{ik}$ for each other dancer P_k that are inversely proportional to the distance between ${}^{\beta}P_i$ and ${}^{\beta}P_k$ as follows:

$$w_{ik} = \max\left(0, \left(1 - \left(\frac{{}^{\beta}P_k - {}^{\beta}P_i}{r}\right)^2\right)^3\right) \tag{3}$$

Here r is a radius of influence, chosen globally for the group. It may be used by the choreographer to enhance the locality of the behavior: a relatively small value of r will prevent a sub-group of nearby dancers from being influenced by a larger sub-group that passes at a distance. The resulting weights are non-zero only within a circle of radius r around ${}^{\beta}P_i$, localizing the effects of LISAMA. This blending kernel is a common polynomial spline [FM03]. Finally, we normalize these weights, dividing each one by their sum.

Let us consider two other frames, a and c, with corresponding time values satisfying $\alpha < \beta < \gamma$. The definition of LISAMA does not impose other constraints on these frames, but in practice, during the first pass of SAMBA, we chose $\alpha = 0$ and $\gamma = 1$. Hence, we compute the approximating choreography from the initial and final frames. In subsequent passes, a and c are selected closer to b, when possible. In the final passes, a, b, and c are three consecutive frames.

We compute an affinity A that maps ${}^{\alpha}P_i$ to ${}^{\gamma}P_i$ and also minimizes the weighted quadratic error for the other dancers. Specifically, we compute a 2×2 matrix L of a linear transformation that minimizes the quadratic norm

$$\sum {}^{\beta}w_{ik}\left(\left({}^{\gamma}P_k - {}^{\gamma}P_i\right) - L\left({}^{\alpha}P_k - {}^{\alpha}P_i\right)\right)^2 \tag{4}$$

This quadratic form may be solved trivially, as discussed in [SMW06, Xie95], but here, instead of the centroids, we use ${}^{\alpha}P_i$ and ${}^{\gamma}P_i$.

Using ${}^{\alpha}V_k$ for ${}^{\alpha}P_k - {}^{\alpha}P_i$ and ${}^{\gamma}V_k$ for ${}^{\gamma}P_k - {}^{\gamma}P_i$, setting the derivative of the above expression to zero yields

$$\sum {}^{\beta}w_{ik}\left({}^{\gamma}V_k - L({}^{\alpha}V_k)\right)\cdot {}^{\alpha}V_k^T = 0 \tag{5}$$

which leads to

$$\sum {}^{\beta}w_{ik}{}^{\gamma}V_k \cdot {}^{\alpha}V_k^T = \sum {}^{\beta}w_{ik}L({}^{\alpha}V_k)\cdot {}^{\alpha}V_k^T \tag{6}$$

and one finally obtains

$$L = \left(\sum {}^{\beta}w_{ik}{}^{\gamma}V_k \cdot {}^{\alpha}V_k\right)\cdot\left(\sum {}^{\beta}w_{ik}{}^{\alpha}V_k \cdot {}^{\alpha}V_k^T\right)^{-1}. \tag{7}$$

We then define the affinity A as the composition of the linear transformation L with a translation by vector ${}^{\gamma}P_i - {}^{\alpha}P_i$. The LISAMA motion from ${}^{\alpha}P_i$ to ${}^{\gamma}P_i$ is $A^t({}^{\alpha}P_i)$. A typical result is shown in figure 5.

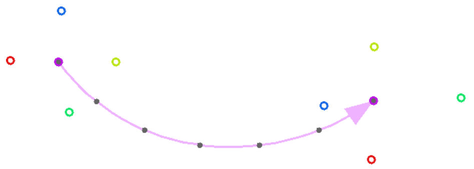

Figure 5: *The steady motion of a given dancer is computed from a modified affine registration or, in cases where no SAM exists, a similarity registration between a preceding frame (left) and succeeding frame (right).*

When the matrix L has a real logarithm, we compute the SAM A^t. When L has no real logarithm, no SAM exists. In this case we find a similarity transform that minimizes (4) as in [SMW06], and use its SAM, which always exists. Both SAMs may be computed using closed form expressions by the EAR algorithm [RV11] or using the approach proposed in [Ale02].

5. SAMBA

The main step of SAMBA computes a target location ${}^{\beta}Q_i$ for a dancer ${}^{\beta}P_i$ of an intermediate frame from the LISAMA of ${}^{\beta}P_i$ as

$${}^{\beta}Q_i = A^u({}^{\alpha}P_i) \tag{8}$$

where the time $u = \frac{\beta-\alpha}{\gamma-\alpha}$.

The first SAMBA pass sets $\alpha = 0$ and $\gamma = 1$ for all intermediate frames and computes these target locations for all dancers in the intermediate frames using ${}^{\beta}P_i = {}^{\beta}Q_i$. Since during this initial pass, we do not have positions for the intermediate frames, we compute weights from the initial and final frames and average them.

Hence, each one of these initial trajectories is steady. A typical result is shown in figure 4.

Algorithm 1

```
for each intermediate frame b {
 a = max(0,b-d);
 c = min(f+1,b+d);
 u = (b-a) / (c-a);
 for each dancer $P_i$ {
   A = LISAMA(i,a,c);
   ${}^{\beta}Q_i$ = $A^u$ (${}^{\alpha}P_i$);
 }
 for each dancer Pi {
   ${}^{\beta}P_i$ = ${}^{\beta}P_i$ + 0.5*(${}^{\beta}Q_i$ - ${}^{\beta}P_i$);
 }
 d = spread(stepnum);
}
```

During each pass, we execute Algorithm 1.

The function `spread` controls a trade-off between local (small spread) and global (large spread) motion prediction. We achieve faster convergence by applying a gradually reducing spread. figure 6 shows a naïve "Fixed Spread" strategy (where the spread is always 1) compared with our preferred "Halving Spread" strategy that starts with the largest possible spread (half the intermediate frames) and reduces this by half at each iteration. Once the spread is reduced to 1 we stop halving and keep a fixed spread until we reach our error threshold and terminate the process.

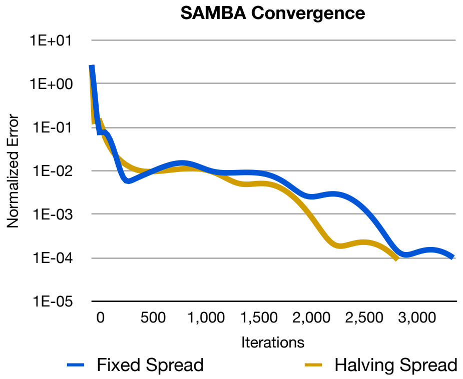

Figure 6: *Error plot for SAMBA iterations using a fixed 1 spread smoothing (blue), and a variable spread smoothing (orange) that repeated halves the spread size.*

Note that this process is similar to a Laplace smoothing of a polygonal curve, except that, instead of using a linear interpolation, we are using LISAMA.

Since steady motions have predictable, stabilizing properties (they vary area monotonically and use minimal acceleration), the resulting smoothing provided by SAMBA inherits these characteristics inasmuch as possible (different dancers may establish contradicting goals for the choreography, and SAMBA must then adopt a compromise, expressed by the kernel radius r, and by making nearby trajectories mutually agreeable.

6. Results and discussion

In this section, we summarize the results of our experiments with SAMBA.

To demonstrate the benefit of subsequent SAMBA passes, we use a configuration where the initial positions of the dancers in the group have two distinct clusters. Although the two clusters are clearly disjoint in both the initial and final frames, their trajectories produced by the first pass of SAMBA result in a temporary "interpenetration" of the two clusters during the choreography. figure 7 shows how subsequent SAMBA passes resolve this interpenetration, delaying the rightward-moving dancers by pushing their trajectories upward and accelerating the leftward-moving dancers at the beginning of their journey.

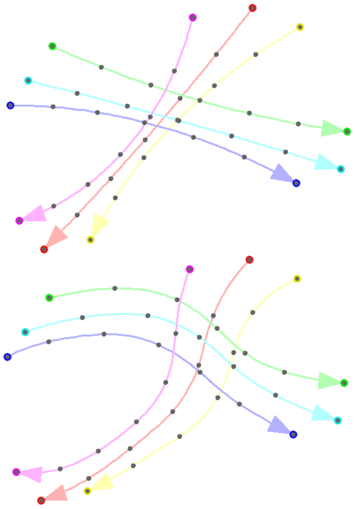

Figure 7: *The first SAMBA pass (top) shows a collision between the two clusters. We show the result after 1000 SAMBA passes (bottom).*

In the accompanying video, we include several example, showing the initial linear choreography and the steadied one produced by SAMBA. Our experiments suggest that SAMBA produces pleasing and interesting choreographies, which are sometimes very different from the linear interpolation.

6.1. Limitations and extensions

SAMBA uses a local search/optimization approach, therefore it may fail to converge to the global optimum or may converge to a solution that the choreographer dislikes.

The choreographer can alter start and end keyframes of the group, but small changes in these initial or final positions may result in qualitatively different overall choreographies. Hence, restricting the choreographer's interaction to the editing of the start and end frames may not provide sufficient flexibility for precise design, even though it may be a valuable automation tool when the choreography is working at a high-level.

Because changes to the initial choreography from which SAMBA starts may lead to qualitatively different solutions, we allow the choreographer to edit the initial solution by adding intermediate constraints to be interpolated by a specific dancer. This is done simply by clicking on a path to pick a dancer i and a time t and by dragging to a new location to establish a constraint for ${}^tP_i$.

For each dancer, we compute a smooth motion that interpolates all such constraints. We use a conjugate gradient method to smoothen the location of the unconstrained positions. The technique works in realtime and the choreographer can directly manipulate the path and see sampled positions. Then, SAMBA is used to steady the resulting motions.

7. Conclusions

We have presented a technique for generating a harmonious choreography of a group of dancers, given only their start and end positions. The technique uses an iterative local smoothing, in the spirit of Laplacian smoothing, estimating dancer motion based on the relative configurations of neighboring dancers. Local neighborhoods are defined by a kernel function with finite support, which enables realtime performance of the SAMBA smoothing.

8. Acknowledgements

References

[ACM*10] ALWANI M., CHAUDHARY R., MATHUR M., ROY S., CHAUDHURY S.: Restricted affine motion compensation in video coding using particle filtering. In *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing* (2010), ACM, pp. 479–484.

[ACOL00] ALEXA M., COHEN-OR D., LEVIN D.: As-rigid-as-possible shape interpolation. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00* (2000), 157–164.

[Ale02] ALEXA M.: Linear combination of transformations. In *SIGGRAPH 2002 Conference Proceedings* (2002), Hughes J., (Ed.), Annual Conference Series, ACM Press/ACM SIGGRAPH, pp. 380–387.

[BA96] BLACK M., ANANDAN P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding 63*, 1 (1996), 75–104.

[BLCD02] BREGLER C., LOEB L., CHUANG E., DESHPANDE H.: Turning to the masters: motion capturing cartoons. *ACM Transactions on Graphics (TOG) 21*, 3 (2002), 399–407.

[Che76] CHEKHOV A. P.: *Complete Works and Letters in Thirty Volumes, Letters*, vol. 8. Nauka, 1976.

[CLRW10] CHAZAL F., LIEUTIER A., ROSSIGNAC J., WHITED B.: Ball map: median projection map between compatible surfaces. *to appear in Int. Journal of Comp. Geometry and Applications* (2010).

[ESE06] EZRA E., SHARIR M., EFRAT A.: On the icp algorithm. In *Proceedings of the twenty-second annual symposium on Computational geometry* (2006), ACM, pp. 95–104.

[FM03] FRIES T., MATTHIES H.: Classification and overview of meshfree methods. *Department of Mathematics and Computer Science, Technical University of Braunschweig* (2003).

[GKB*07] GHOLIPOUR A., KEHTARNAVAZ N., BRIGGS R., DEVOUS M., GOPINATH K.: Brain functional localization: A survey of image registration techniques. *Medical Imaging, IEEE Transactions on 26*, 4 (2007), 427–451.

[Hog53] HOGARTH W.: *The Analysis of Beauty*. Printed by J. Reeves for the author, 1753.

[IMH05] IGARASHI T., MOSCOVICH T., HUGHES J. F.: As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics 24*, 3 (July 2005), 1134–1141.

[JCP*10] JU E., CHOI M. G., PARK M., LEE J., LEE K. H., TAKAHASHI S.: Morphable crowds. *ACM Trans. Graph. 29* (December 2010), 140:1–140:10.

[KC96] KRUGER S., CALWAY A.: A multiresolution frequency domain method for estimating affine motion parameters. In *Image Processing, 1996. Proceedings., International Conference on* (1996), vol. 1, IEEE, pp. 113–116.

[KC98] KRUGER S., CALWAY A.: Image registration using multiresolution frequency domain correlation. In *Proc. British Machine Vision Conf* (1998), Citeseer, pp. 316–325.

[KLLT08] KWON T., LEE K. H., LEE J., TAKAHASHI S.: Group motion editing. In *ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 80:1–80:8.

[Kor02] KORT A.: Computer aided inbetweening. In *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering* (New York, NY, USA, 2002), NPAR '02, ACM, pp. 125–132.

[KR92] KAUL A., ROSSIGNAC J.: Solid-interpolating deformations: construction and animation of pips. *Computers & graphics 16*, 1 (1992), 107–115.

[Kru98] KRUGER S.: *Motion analysis and estimation using multiresolution affine models*. PhD thesis, University of Bristol, 1998.

[MHTG05] MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Meshless deformations based on shape matching. *ACM Transactions on Graphics (TOG) 24*, 3 (2005), 471–478.

[Mod04] MODERSITZKI J.: *Numerical methods for image registration*. Oxford University Press, USA, 2004.

[MSKG05] MÜLLER M., SOLENTHALER B., KEISER R., GROSS M.: Particle-based fluid-fluid interaction. In *ACM SIGGRAPH /Eurographics Symposium on Computer Animation* (Los Angeles, California, 2005), Terzopoulos D., Zordan V., (Eds.), Eurographics Association, pp. 237–244.

[MV98] MAINTZ J., VIERGEVER M.: A survey of medical image registration. *Medical image analysis 2*, 1 (1998), 1–36.

[Rey87] REYNOLDS C.: Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH Computer Graphics* (1987), vol. 21, ACM, pp. 25–34.

[RJ07] RIVERS A. R., JAMES D. L.: FastLSM: fast lattice shape matching for robust real-time deformation. *ACM Transactions on Graphics 26*, 3 (July 2007), 82:1–82:??

[RV11] ROSSIGNAC J., VINACUA A.: Steady affine motions and morphs. *to appear in ACM Transactions on Graphics (TOG)* (2011).

[SDC09] SÝKORA D., DINGLIANA J., COLLINS S.: As-rigid-as-possible image registration for hand-drawn cartoon animations. In *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering* (New York, NY, USA, 2009), NPAR '09, ACM, pp. 25–33.

[SG92] SEDERBERG T., GREENWOOD E.: A physically based approach to 2–D shape blending. *ACM SIGGRAPH Computer Graphics 26*, 2 (1992), 25–34.

[SG01] SURAZHSKY V., GOTSMAN C.: Controllable morphing of compatible planar triangulations. *ACM Transactions on Graphics 20*, 4 (2001), 203–231.

[SGWM93] SEDERBERG T., GAO P., WANG G., MU H.: 2-D shape blending: an intrinsic solution to the vertex path problem. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (1993), ACM, p. 18.

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *ACM Transactions on Graphics (TOG) 25*, 3 (2006), 540.

[TCP06] TREUILLE A., COOPER S., POPOVIĆ Z.: Continuum crowds. In *ACM SIGGRAPH 2006 Papers* (2006), ACM, pp. 1160–1168.

[THL*04] THALMANN D., HERY C., LIPPMAN S., ONO H., REGELOUS S., SUTTON D.: Crowd and group animation. In *ACM SIGGRAPH 2004 Course Notes* (New York, NY, USA, 2004), SIGGRAPH '04, ACM.

[TJ95] THOMAS F., JOHNSTON O.: *The Illusion of Life: Disney Animation*. Disney Press, October 1995.

[TT98] TONER J., TU Y.: Flocks, herds, and schools: A quantitative theory of flocking. *Physical Review E 58*, 4 (1998), 4828.

[WNS*10] WHITED B., NORIS G., SIMMONS M., SUMNER R. W., GROSS M., ROSSIGNAC J.: BetweenIT: An Interactive Tool for Tight Inbetweening. *Eurographics 29*, 2 (2010).

[WR09] WHITED B., ROSSIGNAC J.: B-morphs between b-compatible curves in the plane. In *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling* (2009), ACM, pp. 187–198.

[WR10] WHITED B., ROSSIGNAC J.: Ball-morph: Definition, implementation and comparative evaluation. *IEEE Transactions on Visualization and Computer Graphics* (2010).

[Xie95] XIE M.: Feature matching and affine transformation for 2D cell animation. *The Visual Computer 11*, 8 (1995), 419–428. ISSN 0178-2789.