Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2008)
M. Gross and D. James (Editors)

Staggered Poses: A Character Motion Representation for Detail–Preserving Editing of Pose and Coordinated Timing

Patrick Coleman,[†1] Jacobo Bibliowicz,[1] Karan Singh,[1] and Michael Gleicher[2]

[1]University of Toronto
[2]University of Wisconsin, Madison

Abstract

We introduce staggered poses—a representation of character motion that explicitly encodes coordinated timing among movement features in different parts of a character's body. This representation allows us to provide sparse, pose–based controls for editing motion that preserve existing movement detail, and we describe how to edit coordinated timing among extrema in these controls for stylistic editing. The staggered pose representation supports the editing of new motion by generalizing keyframe–based workflows to retain high–level control after local timing and transition splines have been created. For densely–sampled motion such as motion capture data, we present an algorithm that creates a staggered pose representation by locating coordinated movement features and modeling motion detail using splines and displacement maps. These techniques, taken together, enable feature–based keyframe editing of dense motion data.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Animation

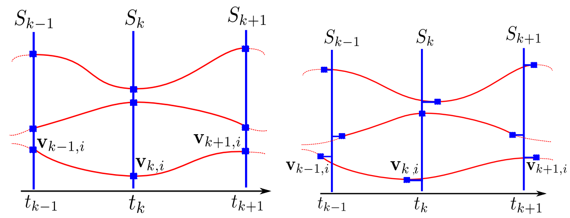
1. Introduction

An important part of a character's motion is the timing that specifies how a character moves from one pose to another. For example, speed of transition is important for conveying mass or character thought [Las87,Las95]. Simply interpolating from pose to pose appears over–constrained and unnatural. Animators have long observed that a motion begins at the hips and propagates out to the extremities [Las87]. To quote Disney, "Things don't come to a stop all at once, guys; first there is one part, and then another" ([TJ81], p.59). In addition, coordinated movement in different limbs is not exactly aligned in time; *non–symmetrical action* is another important part of believable motion [Com99]. Despite the importance of timing variation in modeling believable movement, current animation and motion editing tools provide no support for working with these relationships as a whole.

Traditional poses represent the geometry of the character and are interpolated to create movement. Tools for editing such geometric poses require a large number of similar poses to capture timing variation. If any significant geometric change needs to be made, each of the similar poses created to model the timing variation must be updated. Spline editors and other knot–editing tools allow users to create timing variation at a low level by moving individual knots in the time domain, but making these changes comes with the cost of losing the ability to geometrically alter coordinated extrema as a single pose. Again, users must work across multiple frames to introduce any significant geometric change. In this paper, we introduce *staggered poses*, a representation of motion that explicitly encodes timing variation among coordinated motion extrema; this enables geometric pose–based editing of coordinated extrema that have varied timing as well as the procedural editing of these timing relationships.

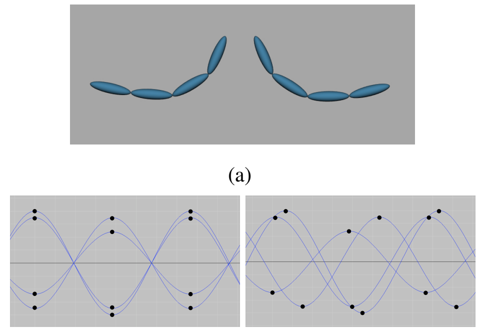
Staggered poses generalize traditional poses by allowing key values of different degrees of freedom to be specified at slightly different times. Relative to a traditional pose (Figure 1a), this representation staggers the timing, using explicitly encoded *timing refinements* (Figure 1b). The timing relationships among these key values determine how the character will pass through the extreme values of the pose and

† {patrick|jacky|karan}@dgp.toronto.edu, gleicher@cs.wisc.edu



(a) (b)

Figure 1: *Pose–based editing systems align all coordinated knots in time (a). Staggered poses (b) incorporate a timing refinement for each knot that encodes timing variation.*



(a)

(b) (c)

Figure 2: *When refining motion from a sequence of rough poses (b) to add timing variation (c), coordination becomes visually confusing in low–level spline–based tools. Even this simple oscillating system with four degrees of freedom (a) quickly becomes disorganized when timing is refined.*

are important for modeling believable propagation of force and intention through a body.

In keyframe animation that represents motion as interpolation between sequential poses, each pose specifies one knot for each degree of freedom, and splines typically specify the interpolated values between the knots. In practice, poses are not often explicitly represented, as animators need to break the temporal alignment of knots to create believable motion. In typical workflows, animators will quickly create poses and block them out in time to sketch out a motion (Figure 2b). The lack of explicit representation of timing refinement among the coordinated knots in each pose has a profound effect on animator workflows. From an interface standpoint, it is difficult to even locate coordinated extreme values, even for simple systems (Figure 2c). This challenge grows in complexity as the number of controls increases. In addition, adding this timing variation eliminates the presence of a representative extreme pose at any given time, causing the coordinated extreme values to no longer be editable as a single geometric entity. Staggered poses, by encoding explicit timing relationships among coordinated extreme values, do allow for geometric pose–based editing of the coordinated extrema *after timing has been refined*, while retaining the mutual coordination among the knots that is important for representing believable transfer of motion from body part to body part.

As in keyframe animation, we use splines to interpolate between sequential extreme values and to provide concise user control over transitions. These splines are defined on a normalized time domain between the refined times of the bounding staggered poses (red, in Figure 3a). As we wish to support the representation of motion with arbitrary high–frequency detail, we also include a displacement map that represents an offset relative to the value interpolated by the spline (green, in Figure 3b). This facilitates the use of staggered poses as a high–level interface for editing coordinated extreme values in densely–sampled data such as recorded motion capture data. This also generalizes layered displacement maps [BW95, WP95]; each temporal region between a pair of adjacent knots can be "interpolated" using a stack of concatenated splines and displacement maps. As interpolation does not exactly meet environmental contact constraints, a final displacement map represents an inverse kinematic solution that modifies the value to meet any such constraints. To facilitate direct geometric pose editing in the presence of timing refinements, we introduce an additional knot at the time of the pose (Figure 3b); this extra knot and the extreme value knot are mutually constrained during editing to maintain continuity.

Overview: We have developed a prototype motion editing system for articulated character motion that uses a motion representation based on staggered poses as its foundation. Users can create motion from scratch, or they can work with motion captured data that is uniformly and densely sampled. Pose–based editing is accomplished by making geometric changes to coordinated extrema as a single unit using existing forward kinematic or inverse kinematic posing tools. We allow them to edit coordinated pose extrema directly, which have slightly different times, or they can directly edit the true pose at the staggered pose time. Our system mutually constrains extreme value knots and editing knots to maintain local continuity. For stylistic control of timing, we provide tools for adding procedural timing variation using a combination of random values and succession patterns. These stylistic timing controls are important, as animators often exaggerate such timing [WH81]. To support the editing of dense articulated character motion data in this keyframe–like setting, we have developed an algorithm that locates coordinated extrema in human motion. It combines such extrema via temporal filtering to create a pose probability signal that indicates the presence of extreme poses; we use these probabilities in conjunction with a local feature measure based on curvature to create staggered pose motion representations that include splines and displacement maps for modeling transitions.

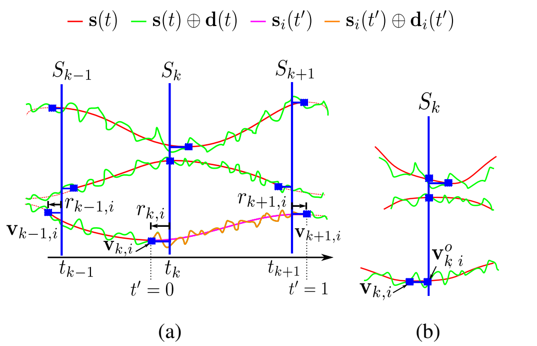


Figure 3: *a) The staggered pose representation models transitions as a composition of splines (red) and displacement maps (green) to provide low–frequency editing controls that preserve detail. b) To facilitate direct geometric editing of poses, we include an additional knot at the pose time; this knot and the knot at the extreme value are mutually constrained to maintain continuity.*

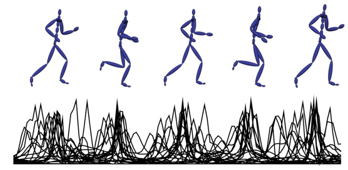


Figure 4: *In recorded human motion, local motion features are temporally correlated to changes in movement of the corresponding body parts. When many parts of the body change direction together, many features cluster nearby in time, which appears as greater peak density and height.*

1.1. Coordinated Features in Human Motion

When abstracting a motion as a sequences of poses, pose times are chosen to correspond to perceptual events of significant motion change. Animators refer to such poses as *extreme poses* and use these poses to specify, conceptually, the content of a motion (See Williams [Wil01], page 65). These extreme poses, which are also often used to lay out a first pass of a new motion, contain the basic information that specifies the geometric aspects of movement. In the cognitive science literature, it has been found that low–level event perception in simple motion is correlated to a number of measurable motion features, most notably peaks of acceleration and other differential measures [Zac04]. In human motion, these features are often coordinated among different parts. We have experimented with a variety of motion features, including maxima of acceleration, directional acceleration, and a scaled variant of directional acceleration, which we define in Section 4. We use the latter measure, although any of these are effective. Figure 4 illustrates plots of this measure for different degrees of freedom in a running motion. The peaks of these plots are directional acceleration features and indicate the presence of direction change. The coordination among the extrema is apparent in the clustering of the peaks; the local spread indicates timing variation among coordinated body parts. Character stance near the center of these clusters, overlaid in Figure 4, is similar to that in extreme poses used by animators. We have found that motion with large intentional movement or physical contacts exhibits the greatest degree of feature coordination and that passive resting motion such as ambient change in a static stance has less–prevalent features. This is not surprising, as resting motion involves significantly smaller forces.

As we use staggered poses to edit existing dense motion data, we need to choose pose times representative of extreme stance and locate existing timing variation. Existing pose selection algorithms based on the minimization of global error [LT01, ACCO05, BZOP07], while good at selecting significantly different poses appropriate for illustration, can create poses where coordinated features do not exist and the motion is smooth. Choosing a good set of staggered poses for geometric editing of existing motion is a similar problem to choosing control points for geometric representation of curves or surfaces. Similarly to how important curve and surface control points are located at geometric features, we use coordinated motion features to locate staggered poses. As coordinated motion features have varied timing, we can explicitly model them with the staggered pose representation. Our algorithm for creating staggered pose representations searches for times with many nearby features to create coordinated staggered pose extrema. It then fits splines and displacement maps to represent the transitions between the staggered poses.

Contributions: The core contribution of this paper is a new representation for motion that explicitly represents relative timing relationships among coordinated degrees of freedom (§3). The key advantage of this representation is that it enables new editing tools that allow for a sparse set of controls to be used for editing coordinated degrees of freedom with timing variation; we introduce these tools in §5. First, §5.1 describes pose–based editing tools that allow the user to alter geometric aspects of motion without damaging either the temporal relationships among body parts or motion details. Second, §5.2 introduces temporal editing tools that alter the timing relationships among degrees of freedom as a coordinated unit, allowing for procedural control over some stylistic aspects of movement. In §4, we provide a “reverse–engineering” method that produces staggered pose representations from existing motion, including recorded motion capture data. Together, these new tools allow for more facile geometric motion editing, as they free the user from explicitly maintaining coordinated temporal relationships during pose–based editing, while enabling high–level temporal adjustments for stylistic control.

2. Related Work

Explicit control over pose–based timing has been provided in animation tools. Some techniques allow motion timing to be specified interactively, given a preexisting set of extreme geometric pose values [NCD*00, IMH05]. These methods treat entire poses as a unit, precluding local control over timing relationships. Many systems in the literature provide methods that alter the timing of existing motion [CCZB00, DYP03, NF05, TM04, WDAC06, MPS06, HdSP07]. While these methods inspire our approach by showing how stylistic effects and "higher–level" control can be achieved through timing manipulation, they do not alter the relative relationships among body parts, which limits the types of effects that can be achieved. One particular inspiration is Neff and Fiume's work, which allows target poses for simulation or interpolation to have varied timing among the degrees of freedom in a specific succession pattern [NF03]. Staggered poses explicitly encode a general class of relative timing relationships and allow for both high–level pose–based editing that preserves these relationships and motion edits that procedurally control this relative timing, allowing for a wider range of effects within a keyframe–style workflow.

Tools that "reverse engineer" recorded motion into editable representations, such as keyframes for interpolation, have existed in commercial products, such as *Character-Studio*, for at least a decade. In recent years, the research community has also developed algorithms for extracting key poses for visualization and compression [LT01, HCHY05, ACCO05,BZOP07]. However, recorded motion data is more typically edited by applying spatial deformations as "layers" of displacement maps†. In the research community, this form of editing has been explored in a variety of systems [BW95, UAT95, WP95, LS99]. Staggered poses build on both approaches. We provide a variant of reverse engineering that constructs a motion representation based on staggered poses to provide the advantages of pose–based editing to recorded data. We use displacement maps in a manner similar to Pullen and Bregler [PB02], but to ensure motion details are preserved, and we enforce constraints similarly to Kovar et al. [KSG02].

3. Representing Motion with Staggered Poses

The staggered pose motion representation models motion with a sparse sequence of key events; each of these events is modeled as a staggered pose that captures not only the values of the character's control variables, but also the local timing relationships among them. Geometric editing of the character's changing stance is facilitated by the sparseness of the staggered poses, while stylistic timing refinements are made by adjusting the explicitly–represented timing.

A staggered pose can contain values for an arbitrary set of control variables. In the case of an articulated figure, these parameters are the root position and orientation and the local joint orientations; we use quaternions for orientations to support smooth spline interpolation. To simplify notation in the remainder of this paper, we define transformation concatenation and factorization operators $\oplus$ and $\ominus$. These are equivalent to vector addition and subtraction for translations and quaternion multiplication and factorization for rotations; note that the latter are non–commutative. By convention, $\mathbf{a} \oplus \mathbf{b}$ applies $\mathbf{b}$ after $\mathbf{a}$, and $\mathbf{c} \ominus \mathbf{a}$ provides the transformation $\mathbf{b}$ such that $\mathbf{a} \oplus \mathbf{b} = \mathbf{c}$.

Motion is represented as a sequence of staggered poses S_k with times t_k. For each staggered pose, each control variable i has a value $\mathbf{v}_{k,i}$ and timing refinement $r_{k,i}$. The equivalent knot in a traditional spline representation is $(t_k + r_{k,i}, \mathbf{v}_{k,i})$. We show these staggered poses in blue in Figure 3a.

To support interpolation control, while retaining the ability to model arbitrary motion detail, we use splines (cubic in our current system) and displacement maps (Figure 3a). We define these on a normalized time domain to provide intrinsic localized time warping when the timing parameters of the poses are changed. To evaluate a control variable $\mathbf{v}_i(t)$ at time t, we identify the neighboring staggered poses S_k and S_{k+1}, as well as the spline $\mathbf{s}_i(t')$ and displacement map $\mathbf{d}_i(t')$ connecting S_k and S_{k+1}. Evaluation is then the composition $\mathbf{v}_i(t) = \mathbf{s}_i(t') \oplus \mathbf{d}_i(t')$, where $t' = (t - (t_k + r_{k,i}))/((t_{k+1} + r_{k+1,i}) - (t_k + r_{k,i}))$, the normalized time between S_k and S_{k+1} for control variable i. This generalizes to multiple splines and displacement maps to support layered representations; note that layer order is important for rotations.

We refer to the character configuration obtained by using extreme values for each variable in a pose as the *non–staggered pose*. Note that the non–staggered pose will never actually exist in the motion, unless all of the timing refinements $r_{k,i}$ are zero. This is different from the *pose at the staggered pose time*, which is obtained by performing interpolation to compute the values for all variables at time t_k. Conceptually, a non–staggered pose often appears as an exaggerated version of the pose at the staggered pose time, as it brings all extrema into temporal correlation. The control variable values of the pose at the staggered pose time are generally convex in value relative to the extrema, resulting in a less extreme appearance.

To support direct editing of the true pose, we include, for each knot where $r_{k,i} \neq 0$, an additional *editing knot* $\mathbf{v}^0_{k,i}$ at the pose time t_k (Figure 3b) . In addition to mutually constraining the relative timing and values of the editing knot and the knot representing the extremum, we also constrain the intermediate spline tangents for continuity. Our tools include two approaches for geometric pose editing; each approach directly manipulates a different knot, and the change is mapped to the other knot. Retiming tools can change the timing refinement value; as the editing knot has no timing re-

† Also known as motion warps.

finement, we instead change its value to maintain continuity. Details are described in §5.

Spline interpolation of sparse poses cannot represent arbitrary detail. This detail is important when editing pre–existing motion or for the potential development of algorithms that create local displacement texture. In addition, motion constraints such as foot plants cannot be modeled using interpolation without a dense sampling of poses. The displacement maps are capable of capturing all these forms of detail. We model constraints on labeled end effectors with associated joint chains. The constraint enforcement displacement map values for each control variable on which such constraints depend are nonzero only for times during which a constraint is active. To avoid discontinuities, we also include a smoothly–attenuated constraint solution in a small window adjacent in time. This provides a mechanism for blending forward and inverse kinematic control. Our localized use of IK constraint enforcement is important, as IK solutions produce notoriously over–correlated motion when used outside a constraint setting. Nearly all solvers ignore neighboring values in time and tend to spread joint adjustment out with a per–joint magnitude that is proportional to the distance of the constraint to a default end effector position. As rotation magnitude is correlated among the joints, IK produces motion extrema that are also correlated in time.

4. Staggered Poses in Dense Motion Data

To support the use of staggered poses for pose–based editing of existing motion data, we have developed an algorithm that locates staggered poses and creates a staggered pose motion representation. Our algorithm takes a top–down approach, first locating the staggered poses, then fitting splines, and finally measuring displacement maps that capture remaining motion detail. As the degree of editing fidelity needed can vary based on the particular editing problem, we prioritize these poses using a heuristic measure and allow users to optionally cull the set of poses to produce fewer editing controls. Location of environmental contact constraints is independent. Figure 4 includes poses at the times chosen using this algorithm, which correspond to large–scale change in movement due to contact and release with the ground.

4.1. Determining Pose Times and Refinements

To locate a sequence of prioritized candidate pose times, we explicitly look for coordinated extrema and heuristically prioritize the perceptual magnitudes of the motion features associated with these extrema. We formulate this probabilistically to allow for the incorporation of different measures, each of which is expressed as a probability $p_i(t)$. We measure a weighted combination of these probabilities, biasing the motion of limbs with significant movement. We apply a low pass filter f to combine peaks close in time, creating an aggregate *pose probability signal* $P(t_k)$. We also include global evidence measures such as changes in labeled constraints as further probability signals, as these indicate the likely presence of a pose. Maxima of the pose probability signal are indicative of strong evidence for correlated extrema and thus indicate a good time for a pose to be created. The peaks of individual pose probabilities $p_i(t)$ are indicative of local knot times; temporal differences between these times and the pose time lead directly to refinement values. To prioritize poses, we bias them by both $P(t_k)$ and the distance to neighboring peaks. We illustrate these ideas in Figure 5.

Local Evidence Measures: While different local evidence measures can be used, we curently use the scaled curvature of local space motion trails. For a single rotation, we create a motion trail in local space $\mathbf{m}_1(j,t)$ that traces the endpoint of a normalized axis toward each child joint j. As this is incapable of encoding twist about the limb, we create a second motion trail $\mathbf{m}_2(j,t)$, using a normalized axis orthogonal to the limb axis. Significant changes in the motion of the limb create regions of high curvature in these motion trails.

We base our measurement of rotational curvature $\kappa(\mathbf{m}(t))$ on the formulae given by Taubin [Tau95]. The discrete measure of rotational curvature is $\kappa_i = 2 * \mathbf{n}_i \cdot \mathbf{e}_i/||\mathbf{e}_i||^2$, given a normal vector $\mathbf{n}_i$ of a Frenet frame and an edge vector $\mathbf{e}_i$ between two samples i and $i+1$. This is the approximation relative to one edge; we weight this measure as computed using both adjacent edges by the corresponding edge lengths. This discrete measure has a range of $[0,\infty)$. We use a modified measure: $\hat{\kappa}_i = \mathbf{n}_i \cdot \mathbf{e}_i/||\mathbf{e}_i||$, which returns a valid probability range of $[0,1]$ and preserves the extrema of curvature for densely sampled data. In addition, this avoids potential problems when minor joints make tight changes in rotation. The true measure of curvature can have extremely large values in tight rotations. Weighting cannot predictably attenuate these values relative to even slightly less sudden rotation changes in other joints that have significantly stronger perceptual motion. The modified measure effectively clamps the extrema to avoid arbitrarily large values.

Given multiple child joints j and two motion trails for each joint, we measure the local probability of a rotation signal as $p(t) = max_{i\in[1,2],j}(\hat{\kappa}(\mathbf{m}_i(j,t)))$. For the translation signal at the root, we use a single motion trail that traces the root position. Finally, as discrete measures are sensitive to noise, we apply iterated Laplacian smoothing; this removes high–frequency noise while preserving the lower–frequency motion content. Parameters for this smoothing are dependent on the source of the motion; we set these once for a given collection by manual adjustment until observed plots of the curvature signals stabilize.

Global Evidence Measures: We use change in environmental contact constraints as additional evidence for pose presence. The start and end times of constraints, such as the moment of contact or release, are indicative of extreme poses. At these times, the discontinuous change in contact has a corresponding acceleration that propagates through the

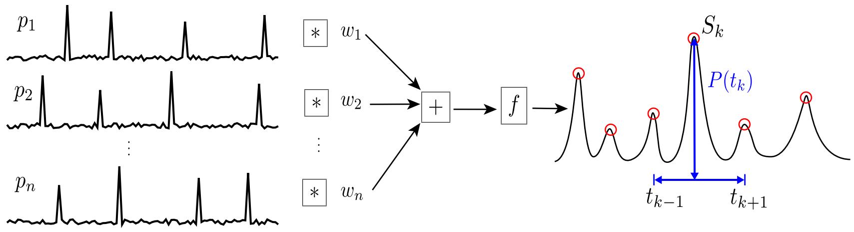


Figure 5: *To locate staggered pose times, we combine signals that measure motion features, filter the result to combine close peaks into an overall probability, and select times at the peaks of this signal, prioritizing by probability and temporal locality.*

body. We model this as a binary signal that is one only at times that constraints become active or disactive.

Aggregating Evidence: We combine our local and global probability metrics to create the aggregate pose probability signal by taking a weighted sum of probability signals and filtering. Weighting the results allows us to positively bias signals associated with limbs that are moving most significantly in space. Applying a low–pass filter allows peaks of probability to positively bias the overall probability of adjacent frames; this has an effect of combining closely–clustered peaks of overall probability into a single peak. Given a set of pose probability measures $p_i(t)$, the overall pose probability is $P(t) = f(\sum_i w_i p_i(t) / \sum_i w_i)$, where f is a low–pass tent filter; we use a filter width of 0.15 seconds.

Rotation signal weight values consider limb length and motion magnitude, we set $w_i = \sum_j |\mathbf{x}_i - \mathbf{x}_j||\mathbf{m}_j|$; $\mathbf{x}_i$ is the joint position in world space and $\mathbf{x}_j$ is the world space position of child joint j. $|\mathbf{m}_j|$ is the arc length of the motion trail $\mathbf{m}_j$. For constraints, we set w_i to a nonzero value when change in constraint state is indicative of extreme poses.

Prioritized Pose Extraction: Candidate poses are created at all local maxima of the pose probability signal. We force the first and last frame of the motion to have poses, as this is necessary to fully reconstruct the motion. Physical motion tends to have very sparse poses, as muscle activations are correlated with contact events. Looser motion often has more candidate poses, although it remains a sparse set. We provide users with the option of culling this set interactively. To favor both regular temporal sampling and poses with high probability, we define a weight w_k, for each pose S_k. This weight is $w_k = P(t_k)(t_{k+1} - t_{k-1})$; the second half of the term positively biases poses with more distant neighbors. We sort these weights w_k and display the highest–weighted percentage of the candidate poses as specified interactively by the user. Users can adjust a slider to select a desired number of high priority poses, specify a time interval of interest from which to choose poses, or individually select from the candidate set.

Timing Refinements: We set refinements by searching for local extrema near the pose time. We only consider those closer to the given pose than neighboring poses, and we bound these refinements in time by 0.25 seconds relative to the pose time. If no extremum exists, we set the refinement to zero. This temporal bound has been empirically set; all natural human motion that we have analyzed has a cluster width of about 0.5 seconds.

4.1.1. Fitting Splines and Displacements

Given a sequence of staggered poses and times (S_k, t_k), we fit spline tangents to the time samples by minimizing squared distance (for translation) or squared angular distance on SO(3) (for rotation). The first problem has a least–squares linear solution. As we use cubic quaternion splines defined using recursive spherical linear interpolation for rotation, the second problem requires a nonlinear optimization (recursive spherical linear interpolation) with nonlinear constraints (unit length quaternions). We simplify the problem to an unconstrained optimization by modeling the constraints as an additional term. We then use a general nonlinear optimization method and renormalize the quaternions. We refer the reader to Appendix A for details.

Given an interpolated spline value $\mathbf{s}_i(t')$ for control variable i at normalized time t', as well as the actual value $\mathbf{v}_i(t')$, displacement measurement is $\mathbf{d}_i(t') = \mathbf{v}_i(t') \ominus \mathbf{s}_i(t')$.

4.1.2. Finding Constraints

A number of algorithms have been developed to detect environmental contact constraints [BB98, IAF06, LB06]. As previously noted, these algorithms perform well, but often make mistakes, and most require manual cleanup [KSG02]. We use a heuristic algorithm in conjunction with interactive constraint editing tools; details are presented in Appendix B.

5. Motion Editing with Staggered Poses

Our motion editing tools allow users to work directly with the staggered pose representation at different levels of fidelity. Geometric manipulation of pose, described in §5.1, allows users to work at the highest level of geometric abstraction, with preservation of coordinated timing. Our temporal editing tools, described in §5.2, allow users to manipulate timing refinements as a coordinated unit for stylistic control. Users can also adjust intermediate control points of the interpolating splines and the values of the displacement map for fine–grained control of transitions.

5.1. Geometric Pose Editing

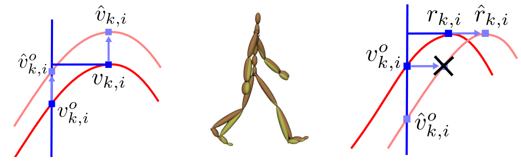
Our geometric pose editing tools allow users to manipulate coordinated motion extrema with timing variation as a single unit. By changing the extreme poses that the character passes through, these tools can be used, for example, to change the style of a character's stance, position the limbs in new configurations, or scale the magnitude of a movement. We provide two approaches to editing coordinated extrema; direct editing of the pose values at the time of the staggered pose and direct editing of all extreme values.

Our first approach to geometric editing allows users to edit the true pose at the staggered pose time. The editing knots $\mathbf{v}^o_{k,i}$ represent these true values, and by mutually constraining them to the extreme value knots, editing changes are applied to the extreme knots as equal changes. This avoids the introduction of a short, large acceleration. To enforce this constraint, the updated extreme value knot is $\hat{\mathbf{v}}_{k,i} = \mathbf{v}_{k,i} \oplus (\hat{\mathbf{v}}^o_{k,i} \ominus \mathbf{v}^o_{k,i})$, given the new value $\hat{\mathbf{v}}^o_{k,i}$ of the editing knot. We illustrate this in Figure 6a.

The second approach is possible with or without the presence of the editing knots $\mathbf{v}^o_{i,k}$. To directly edit all extreme values, we collapse timing refinements to zero, allow the user to edit the non–staggered pose, and then return the refinements to their previous values. The non–staggered pose appears as a slightly exaggerated version of the true pose at the staggered pose time, as any parameters with timing refinements will have slightly more extreme values. We show this in Figure 6b. We offset the values of any editing knots as $\hat{\mathbf{v}}^o_{k,i} = \mathbf{v}^o_{k,i} \oplus (\hat{\mathbf{v}}_{k,i} \ominus \mathbf{v}_{k,i})$ to maintain the mutual value constraint. This approach is useful when coordinated extreme values have meaningful bounds. For example, adjusting a joint near its rotation limit is easier when the extreme value is manipulated, in comparison to the less extreme value of an editing knot.

As many edits will push poses to more extreme or less extreme values, we provide the ability to edit a pose relative to another pose. This can be any pose of the user's choosing, such as a neighboring pose for local exaggeration or an average pose for global exaggeration. The user has control of a weight value w_r that specifies the degree of offset; we update each value as $\hat{\mathbf{v}}_{k,i} = \mathbf{v}_{k,i} \oplus w_r(\mathbf{v}_{k,i} \ominus \mathbf{v}^r_i)$[‡], given the relative pose values $\mathbf{v}^r_i$. As this operates on values, it can be applied to either the pose at the staggered pose time or the non–staggered pose, with values updated as described above.

‡ For orientations, premultiplication by w_r represents a rotation scale rather than component–wise quaternion multiplication.



(a) (b) (c)

Figure 6: *We mutually constrain editing knots and extreme value knots to maintain continuity. Change in value is copied from one to the other (a). The non–staggered pose, shown in orange, appears exaggerated relative to the true pose at the staggered pose time (yellow). Change in timing refinement requires spline evaluation and re–creation of the editing knot to maintain continuity (c).*

If users create or delete poses, we update the splines and displacement maps to maintain the unit–length domain. In general, this will change the motion, as a change in the knots of the transition spline will change interpolation evaluation. To compensate, we update the displacement map to preserve the global transition values.

5.2. Temporal Editing

Our temporal editing tools allow users to specify how timing varies among body parts. The simplest form of temporal editing is retiming staggered poses as a unit; this simply involves updating t_k. This can be effective for changing perceived energy or weight. As refinements, splines, and displacement maps are defined relative to this timing, no other change to the motion representation is needed.

We provide two tools for applying procedural staggered timing; one applies succession–based offsets and the other applies random offsets. As originally presented by Neff and Fiume [NF03], succession follows an incremental profile down the limb. Let each control parameter i have a depth $d(i)$, relative to some user–selected joint. Given a base timing offset s, we can apply this to a staggered pose as the refinement update $\hat{r}_{k,i} = r_{k,i} + sd(i)$. We generalize this to create a new nonlinear succession pattern capable of modeling a greater variety of stylistic movement. Our generalized update is $\hat{r}_{k,i} = r_{k,i} + sd(i)^l$, where l controls how nonlinear the timing is. For $l = 1$, this is the incremental succession of Neff and Fiume. Increasing l causes a limb to appear less constrained, with inertia having a greater apparent effect on the movement; this creates a cartoony feel to the timing. For random offsets, we apply the change in timing as $\hat{r}_{k,i} = \hat{r}_{k,i} + rand(-\tau,\tau)$; τ is a parameter for bounding the random timing. Random offsets are appropriate for modeling asymmetric timing among different limbs; we allow users to select a set of joints and apply the effect only to those joints. Succession and random timing variation can be combined to both stagger the timing of different limbs rel-

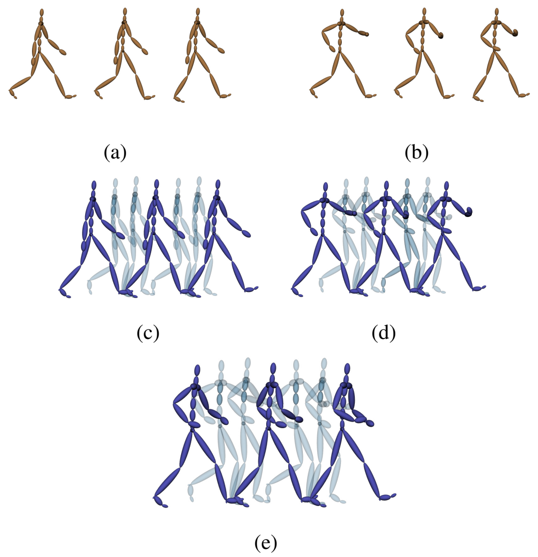


Figure 7: . *In this example, we apply geometric editing to the non–staggered poses (a) to change the shape of the arm swing (b) and nonlinear succession to loosen up the timing. We show the original motion (c), the motion with only geometric pose edits (d), and the motion with both geometric and temporal edits (e).*

ative to each other and to apply a succession pattern down each limb.

As with any retiming algorithm, care must be taken to avoid large nonlinear time warps, as this can introduce sudden acceleration. We warn users when refined times are set large enough to be closer to a neighboring staggered pose than the staggered pose with which the knot is associated.

When editing knots are present, we must update them to maintain local continuity. As shown in Figure 6c, we first move the editing knot to the global time $t_k + (\hat{r}_{k-i} - r_{k,i})$. This, however, is not an appropriate time for direct editing of pose value; therefore, we then return the editing knot to t_k with an updated value. To do this, we first evaluate the splines and displacements at t_k to determine the new editing knot value $\hat{\mathbf{v}}^o_{k,i}$. We then replace the temporally changed editing knot with the new editing knot at t_k with value $\hat{\mathbf{v}}^o_{k,i}$. Similarly, if a procedural timing tool requires a new editing knot to be created, we insert it at t_k by evaluating splines and displacement maps. In each case, we update the displacement map to compensate for changes in spline interpolation.

6. Results and Discussion

The staggered poses motion representation and associated motion editing tools have been implemented within the commercial animation system *Maya*. This allows us to use existing control hierarchies as well as tools that support inverse kinematic positioning to make geometric edits. In comparison to pose–based editing, staggered poses require fewer poses to capture important aspects of timing variation, which allows us to provide sparse geometric controls. In comparison to independent splines, staggered poses can be retimed and changed in geometric value without requiring users to explicitly manage coordinated timing relationships.

We have used our editing tools to edit a number of motions, primarily to introduce stylistic effects, such as pose exaggeration and successive timing. An example edit of a walking motion is demonstrated in Figure 7. In this example, we apply geometric edits to change how the arms are held in the non–staggered poses. We then apply nonlinear succcession to loosen up the arm movement. Subtleties of transition are carried along in the retimed splines and displacement maps. Figure 8 includes two frames from both a carrying motion and a jump. This motion was edited using both the geometric tools and temporal tools interchangeably.

An immediate question is how much effect the use of features has on editing in comparison to general locations in time. Feature–centric control is important, as feature position and timing have a direct effect on how we identify important events in a motion [Zac04]. In addition, users must be cognizant of extreme values, as they often reach believable limits of body shape for a given motion. Editing at points in time away from features provides only indirect control over their value. In addition, any manual change in value at locations away from features is likely to introduce new motion features, as common interfaces locally modify values of splines or displacements. Staggered poses, by explicitly representing the relationship among coordinated extrema, provide concise direct control over the coordinated features. In comparison to motion editing with layers of splines and displacement maps, the staggered poses representation provides all the power of layered editing, but with sparse feature–based controls. Layered editing approaches also require users to specify temporal layering bounds; by smoothly adjusting all values with respect to neighboring features, we do not impose such a requirement.

While we use inverse kinematic tools to adjust poses for geometric editing, we only apply the IK solution to the sparse set of poses used for interactive editing. This is important, as animated IK configurations produce over–correlated motion. Our use of constraints as additional local IK controls is similar to how animators use blends of forward kinematic and IK controls when animating to minimize over–correlation.

When applying temporal edits to recorded human motion, care must be taken during motion processing to avoid artifacts associated with correlated noise in sequential joints. Many motion capture pipelines filter marker positions and fit skeletons to these filtered signals. If sequential joints are nearly linear, as is often the case with the back, fitting algorithms can create small noise contributions to the rotations

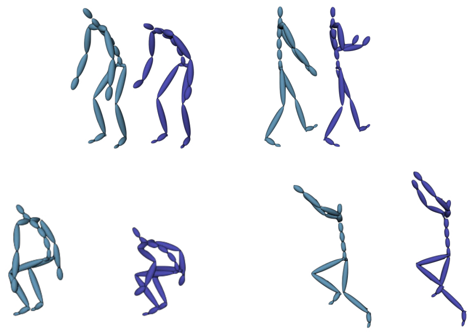


Figure 8: . *Additional editing examples: A combination of geometric and temporal editing creates a stronger anticipation to a lifting movement and changes how an object is held (top). Frames from the original motion are light blue and the edited motion frames are shown in violet. These tools can be used interchangeably, as was done to create a more expressive landing and leap (bottom).*

that can have a perceptual effect on the overall body motion. However, these will correlate with child joints to approximately cancel out on descendent joints, and the effect is usually not perceptible. De–correlating these signals in time removes this cancelation, and short high–frequency artifacts can appear in descendent joints. Appropriate filtering of rotation data before editing avoids this problem [LS02].

One concept that we have not fully addressed in this paper is the use of *partial poses*—these are poses that control only a subset of the degrees of freedom. These can be important for creating independent sets of controls to facilitate editing movement in one part of a body independently of another, another workflow practice used by animators [Las87,Wil01]. Our system does provide support for partial poses, although our algorithm for locating staggered poses does not attempt to partition degrees of freedom.

7. Conclusion and Future Work

In this paper, we have introduced staggered poses, a new motion representation that explicitly encodes timing variation among coordinated degrees of freedom. This allows users to make sparse geometric changes to coordinated extreme values with timing variation and to apply procedural timing variation, while preserving low–level detail of movement. One key benefit of this approach is that pose–centric editing can be applied to refined motion without introducing new undesired movement features and without requiring users to manage coordination in timing variation among many degrees of freedom. A second benefit is that common refinements to timing can be procedurally applied, rather than manually specified through low–level knot manipulation. We have demonstrated the applicability of the staggered pose motion representation to making high–level geometric changes or timing changes to coordinated extreme values in existing motion data, such as recorded motion capture data. To facilitate this, we have introduced an algorithm that locates likely extreme poses with timing variation by searching for coordinated motion features.

As part of our ongoing work, we are investigating how staggered pose representations can be integrated with pose control graphs and motion graphs as concise descriptions of dense motion data. Our prototype system has been implemented within the commercial animation system *Maya*, but our current interface does not yet support interactive editing of low–level components such as splines and displacement maps. We are currently adding these capabilities, as well as providing a better integration with *Maya*'s existing animation toolset; this will allow us to conduct a user study to evaluate the effectiveness of our editing tools with respect to existing techniques and to eventually release the tools for public use. Finally, we are interested in investigating how coordinated timing variation can be used to refine or analyze the motion of simulated deformable objects or facial models.

Acknowledgements: We thank MITACS for financial support and Autodesk for the donation of *Maya*.

Appendix A: Fitting Orientation Splines

Given a sequence of quaternion orientations $\mathbf{q}_i$ at corresponding times t_i, where $t_1 = 0$ and $t_n = 1$, we fit a cubic orientation spline $\mathbf{q}(t)$, where cubic interpolation is defined using recursive spherical linear interpolation [Sho85]. Denote this as $\mathbf{q}(t) = \mathbf{s}(\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2, \hat{\mathbf{q}}_3, \hat{\mathbf{q}}_4, t)$ for control orientations $\hat{\mathbf{q}}_1$, $\hat{\mathbf{q}}_2$, $\hat{\mathbf{q}}_3$, and $\hat{\mathbf{q}}_4$ and time t. For endpoint interpolating splines, $\hat{\mathbf{q}}_1 = \mathbf{q}_1$, $\hat{\mathbf{q}}_4 = \mathbf{q}_n$, and $\hat{\mathbf{q}}_2$ and $\hat{\mathbf{q}}_3$, the "tangent" orientations, are free parameters. We model this problem as a general nonlinear optimization:

$$f(\hat{\mathbf{q}}_2, \hat{\mathbf{q}}_3) = w_f f_f(\hat{\mathbf{q}}_2, \hat{\mathbf{q}}_3) + w_c f_c(\hat{\mathbf{q}}_2, \hat{\mathbf{q}}_3).$$

The first term, f_f adjusts the tangents to fit the spline to the data samples by minimizing the squared angular difference on SO(3):

$$f_f(\hat{\mathbf{q}}_2, \hat{\mathbf{q}}_3) = \frac{1}{n-2} \sum_{k=2}^{n-1} cos^{-1}(\mathbf{q}_k \cdot \mathbf{s}(\mathbf{q}_1, \hat{\mathbf{q}}_2, \hat{\mathbf{q}}_3, \mathbf{q}_n, t_k))^2.$$

The fraction normalizes the sum by the number of samples to maintain independence between w_f and the sample count. The second term models the unit length constraints on quaternions that represent orientations. This is

$$f_c(\hat{\mathbf{q}}_2, \hat{\mathbf{q}}_3) = \left| \|\hat{\mathbf{q}}_2\|^2 + \|\hat{\mathbf{q}}_3\|^2 - 2 \right|.$$

We minimize $f(\hat{\mathbf{q}}_2, \hat{\mathbf{q}}_3)$ using a general nonlinear optimization solver, the Nelder–Mead downhill simplex method [PTVF92]. We use linear rotation splines as the initial solution. This optimization will generally not reach zero, and some error will be contributed by the second term. We therefore renormalize $\hat{\mathbf{q}}_2$ and $\hat{\mathbf{q}}_3$. We have experimented with different values for w_f, our fitting weight, and w_c, our constraint weight. Setting these to values such that $w_f = 1000w_c$ produces quaternions that are close to unity in magnitude and causes the splines to approximate the data well.

Appendix B: Environmental Contact Constraints

Our algorithm for constraint detection is similar in spirit to that of Bindiganavale and Badler [BB98]. We incrementally search forward in time for joints whose world space positions remain still, within a given tolerance, for a given minimum length of time (0.25 seconds). When these conditions are met, we create a constraint and expand it in time to account for further frames that remain within the given positional tolerance. We set our tolerance to be slightly larger than the apparent error introduced by noise, which we measure from example constraints that we manually label.

References

[ACCO05] ASSA J., CASPI Y., COHEN-OR D.: Action Synopsis: Pose Selection and Illustration. *ACM Trans. Graph. 24*, 3 (2005), 667–676.

[BB98] BINDIGANAVALE R., BADLER N. I.: Motion Abstraction and Mapping with Spatial Constraints. In *Proceedings of CAPTECH* (1998), pp. 70–82.

[BW95] BRUDERLIN A., WILLIAMS L.: Motion Signal Processing. In *Proceedings of SIGGRAPH* (1995), pp. 97–104.

[BZOP07] BOUVIER-ZAPPA S., OSTROMOUKHOV V., POULIN P.: Motion Cues for Illustration of Skeletal Motion Capture Data. In *Proceedings of NPAR* (2007), pp. 133–140.

[CCZB00] CHI D., COSTA M., ZHAO L., BADLER N.: The EMOTE Model for Effort and Shape. In *Proceedings of SIGGRAPH* (2000), pp. 173–183.

[Com99] COMET M.: Character Animation: Principals and Practice. *http://www.comet-cartoons.com/3ddocs/charanim/* (1999).

[DYP03] DONTCHEVA M., YNGVE G., POPOVIĆ Z.: Motion Synthesis from Annotations. *ACM Trans. Graph. 22*, 3 (2003).

[HCHY05] HUANG K., CHANG C., HSU Y., YANG S.: Key Probe: A Technique for Animation Keyframe Extraction. *The Visual Computer 21*, 8–10 (2005), 532–541.

[HdSP07] HSU E., DA SILVA M., POPOVIĆ J.: Guided Time Warping for Motion Editing. In *Proceedings of SCA* (2007).

[IAF06] IKEMOTO L., ARIKAN O., FORSYTH D.: Knowing When to Put Your Foot Down. In *Proceedings of I3D* (2006).

[IMH05] IGARASHI T., MOSCOVICH T., HUGHES J. F.: Spatial Keyframing for Performance–Driven Animation. In *Proceedings of SCA* (2005).

[KSG02] KOVAR L., SCHREINER J., GLEICHER M.: Footskate Cleanup for Motion Capture Editing. In *Proc. SCA* (2002).

[Las87] LASSETER J.: Principles of Traditional Animation Applied to 3D Computer Animation. In *Proc. SIGGRAPH* (1987).

[Las95] LASSETER J.: Tricks to Animating Characters with a Computer. *SIGGRAPH Course Notes* (1995).

[LB06] LE CALLENNEC B., BOULIC R.: Robust Kinematic Constraint Detection for Motion Data. In *Proceedings of SCA* (2006).

[LS99] LEE J., SHIN S. Y.: A Hierarchical Approach to Interactive Motion Editing for Human–Like Figures. In *Proc. SIGGRAPH* (1999).

[LS02] LEE J., SHIN S.: General Construction of Time–Domain Filters for Orientation Data. *IEEE Transactions on Visualization and Computer Graphics 8*, 2 (2002), 119–128.

[LT01] LIM I., THALMANN D.: Key–Posture Extraction out of Human Motion Data by Curve Simplification. In *Proceedings of the IEEE Conf. on Engineering in Medicine and Biology* (2001).

[MPS06] MCCANN J., POLLARD N. S., SRINIVASA S.: Physics–Based Motion Retiming. In *Proceedings of SCA* (2006).

[NCD*00] NGO T., CUTRELL D., DANA J., DONALD B., LOEB L., ZHU S.: Accessible Animation and Customizable Graphics via Simplicial Configuration Modeling. In *Proc. SIGGRAPH* (2000).

[NF03] NEFF M., FIUME E.: Aesthetic Edits for Character Animation. In *Proceedings of SCA 2003* (2003), pp. 239–244.

[NF05] NEFF M., FIUME E.: AER: Aesthetic Exploration and Refinement for Expressive Character Animation. In *Proceedings of SCA* (2005).

[PB02] PULLEN K., BREGLER C.: Motion Capture Assisted Animation: Texturing and Synthesis. *ACM Trans. Graph. 21*, 3 (2002), 501–508.

[PTVF92] PRESS W. H., TEUKOLSKY S. A., VETTERLING W. T., FLANNERY B. P.: *Numerical Recipes in C: The Art of Scientific Computing (2nd ed.)*. Cambridge University Press, 1992.

[Sho85] SHOEMAKE K.: Animating Rotation with Quaternion Curves. In *Proceedings of SIGGRAPH* (1985), pp. 245–254.

[Tau95] TAUBIN G.: Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation. In *Proc. ICCV* (1995).

[TJ81] THOMAS F., JOHNSTON O.: *Disney Animation: The Illusion of Life*. Abbeville Press, 1981.

[TM04] TERRA S. C. L., METOYER R. A.: Performance Timing for Keyframe Animation. In *Proceedings of SCA* (2004).

[UAT95] UNUMA M., ANJYO K., TAKEUCHI R.: Fourier Principles for Emotion–Based Human Figure Animation. In *Proceedings of SIGGRAPH* (1995).

[WDAC06] WANG J., DRUCKER S. M., AGRAWALA M., COHEN M. F.: The Cartoon Animation Filter. *ACM Trans. Graph. 25*, 3 (2006).

[WH81] WHITAKER H., HALAS J.: *Timing for Animation*. Focal Press, 1981.

[Wil01] WILLIAMS R.: *The Animator's Survival Kit*. Faber and Faber, 2001.

[WP95] WITKIN A., POPOVÍC Z.: Motion Warping. In *Proceedings of SIGGRAPH* (1995).

[Zac04] ZACKS J.: Using Movement and Intentions to Understand Simple Events. *Cognitive Science* (2004), 979–1008.