Planar similarity-motion interpolating three keyframes: Comparative assessment of prior and novel solutions

Jarek Rossignac* and Àlvar Vinacua

Georgia Tech, Atlanta, USA and UPC Barcelona, Spain

Abstract

We compare 8 solutions for defining the planar motion of an oriented edge that interpolates 3 keyframes. One contribution is the discovery of several novel solutions, one of which produces what we call a ***locally-perseverant*** motion, for which the acceleration of a moving point remains constant in the local (moving) frame. The other contribution is to demonstrate that: (a) many interesting solutions exist, (b) the mathematical and perceived differences between the animations they produce are significant, and (c) these differences may matter for designers and applications. To allow motions that rotate by more than 2π, we represent the 3 keyframes and the moving edge by ***arrows***, each storing the starting-point $\mathbf{p}$ of the edge, its length m, and its ***winding*** (arbitrary angle) w. Hence, an arrow defines an integer winding-count k (with $|w - 2k\pi| \leq \pi$) and a ***similarity*** transformation that combines dilation by m, rotation by $w - 2k\pi$, and translation from the origin to $\mathbf{p}$. Our chosen ***PITA*** (Planar Interpolation of Three Arrows) solutions are formulated using compositions of linear, polar, or log-spiral interpolations, or using ODEs or logarithms of matrices. We compare these solutions in terms of 11 mathematical properties and also in terms of subjective attributes that may be important for designers. We illustrate differences between our 8 chosen PITAs in 6 use-cases: Keyframe-animation, Variable-width stroke design, Banner deformation, Pattern animation, Motion prediction, and Curve design.

Keywords: Planar Animation, Shape-Preserving Motion, Interpolation of 3 Similarities, Interpolation of 3 Edges

1. Introduction

The term ***similarity*** refers here to an affine transformation in the plane (not just its linear part) and the term ***motion*** refers to a time-parameterized similarity.

We discuss motions that interpolate 3 ***timed-keyframes*** (similarities associated each with a different time-value). We present: In Sec. 2, the concepts of winding, wector, arrow, and twirl (which we introduce to remove discontinuities during animations and popping during editing and to support looping motions) and a tutorial on steady motions and morphs; In Sec. 3, the formulations of our 8 chosen solutions; In Sec. 4, an objective comparison of these solutions using 11 mathematical properties that may be important for applications; In Sec. 5, the relation of our PITAs to prior art. In Sec. 6, suggestions for measuring goodness and assessing ease-of-use; In Sec. 7, demonstrations of differences between our chosen solutions in 6 use-examples; The mathematical details of some of these 8 chosen solutions, their advantages over many other options, and the broader prior art are discussed in appendices.

We model a similarity using an ***arrow*** represented by its origin, angle, and magnitude. Hence, the problem may be formulated as follows. Given 3 coplanar keyframe arrows, $\mathbb{A}$, $\mathbb{B}$, and $\mathbb{C}$, and 3 associated time-values, a, b, and c, define a smooth similarity-motion (represented by an arrow-valued function of time), $\mathbb{D}(t)$, that satisfies the 3 interpolation constraints: $\mathbb{D}(a) = \mathbb{A}$, $\mathbb{D}(b) = \mathbb{B}$, and $\mathbb{D}(c) = \mathbb{C}$. We compare 8 chosen solutions (Fig. 1) denoted P1 through P8. We call them ***PITA***s (Planar Interpolation of Three Arrows). Abbreviations Pxy stands for Px and Py, Pxyz for Px, Py, and Pz, etc. For aesthetic or functional reasons, the differences between the animations produced by these solutions may be important in some applications.

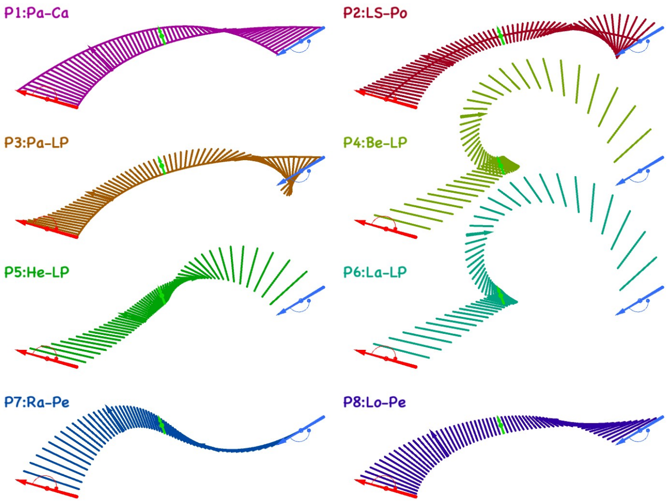


Figure 1: 8 PITAs interpolating the same 3 timed-keyframes (red, green, blue) with even timing, $b = (a+c)/2$. We visualize $\mathbb{A}$, $\mathbb{B}$, and $\mathbb{C}$ as arrows and the n intermediate frames $\mathbb{D}(a + (c-a)\frac{i}{n})$ as line-segments. P178 are not aware of the winding, which is shown as a spiral around the keyframe's midpoint here (or around its startpoint in other figures). We draw the trajectories of the two endpoints (for P1), of the midpoint (for P2), and of the startpoint (for P3), which are all independent of the direction and magnitudes of the keyframes.

[*] Corresponding author

Email address: `jarek@cc.gatech.edu, alvar@cs.upc.edu` (Jarek Rossignac* and Àlvar Vinacua)

The challenge is not to produce a mathematically valid solution with sufficient continuity. Indeed, as demonstrated here, many different solutions exist. We posit that the true challenge is to choose a small set of solutions so that: (1) for each configuration of constraints, one or more of them produce an "acceptable" result; (2) the resulting animation of an object $\mathbf{O}$ is independent of the choice of the origin and orientation, and of the placement of $\mathbb{A}$ with respect to $\mathbf{O}$; (3) the designer is able to "roughly predict" the effect, on the animation, of tweaking any one of the parameters (origin, angle, length) of any of the keyframe arrows; and (4) the ***gain*** (ratio of the magnitude of the change in the animation that results from such a tweak to the magnitude of the tweak) is not "surprisingly large", at least for a "sufficiently broad" set of configurations. This challenge is further complicated by the fact that we may not have mathematical formulations for the terms in quotes used above. Most prior art focuses on the mathematical formulation of a small set of solutions, shows only few examples, which avoid large rotations between keyframes, and does not assess formal or subjective properties of the proposed solutions. Hence, further work is needed to validate and compare solutions. This is particularly difficult in 3D. Hence, before attempting this complex assessment-challenge in 3D, we focus here on 2D solutions, for which realtime visualization and direct manipulation help with empirical validation and comparison.

2. Winding, Arrow, Twirl, and Steady Similarities

In this section, we define our terminology and notation for the basic concepts used in the paper and we explain how we represent and compute them. In particular, we introduce the novel concepts of winding, wector, arrow, twirl, and steady morph, average, and transport of arrows, which we combine to define twirling interpolations, as the one in Fig. 2.

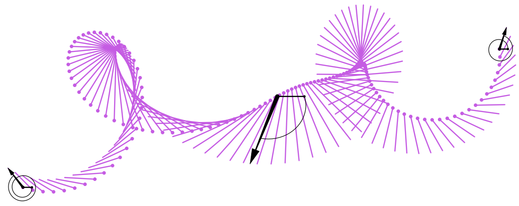


Figure 2: Frames of the motion of an arrow that interpolates three keyframes (black), with windings shown as spirals around their origin.

Technical terms (whether standard or introduced) are in ***bold-italics*** when first mentioned and when defined. Stressed words are underlined. We use "***iff***" for "if and only if".

2.1. Numbers, Points, Vectors, Arrows

Integer ***counts*** and ***indices*** are shown in lowercase (i).

Reals (time, angle, ratio) are in lowercase italics (w).

$\mathbf{p}$ denotes a ***point***. Constructor $\mathrm{P}(x, y)$ returns a point with Cartesian coordinates x and y.

$\vec{\mathbf{v}}$ denotes a vector. Constructor $\mathrm{V}(x, y)$ returns a vector with Cartesian components x and y. $|\overrightarrow{\mathbf{ac}}|$ is the distance between $\mathbf{a}$ and $\mathbf{c}$. Constructor $\mathrm{V}_{\mathrm{pol}}(m, w)$ returns $\mathrm{V}(m\cos w, m\sin w)$, with $m = |\vec{\mathbf{v}}|$ and $w = \mathrm{angle}(\vec{\mathbf{e}}, \vec{\mathbf{v}})$, where $\vec{\mathbf{e}} = \mathrm{V}(1, 0)$. $\vec{\mathbf{u}} \bullet \vec{\mathbf{v}}$ denotes a dot-product.

Wectors generalize the polar form (m, w) of a vector by differentiating between all trigonometric branches of angle (argument) w, which we call ***winding***. w needs not lie in $[-\pi,\pi]$. When $w \in [-\pi, \pi]$, the wector is said to be ***zero-turn***. We use the vector notation for wectors. When used in $\mathbf{p}+\vec{\mathbf{w}}$ expressions or in a context where zero-turn is assumed, a wector is treated as the corresponding zero-turn vector.

Arrows are in uppercase outline ($\mathbb{A}$). Arrow from point $\mathbf{p}$, with magnitude m and winding w is written $\mathbb{A}_{\mathbf{p},m,w}$ and constructed using $\mathrm{A}(\mathbf{p}, m, w)$. Arrow from point $\mathbf{p}$, with the magnitude and winding of wector $\vec{\mathbf{w}}$ is written $\mathbb{A}_{\mathbf{p},\vec{\mathbf{w}}}$ and constructed using $\mathrm{A}(\mathbf{p}, \vec{\mathbf{w}})$. We use the following notation to access its parameters: ***origin*** $\mathbb{A}.\mathbf{p}$, ***magnitude*** $\mathbb{A}.m$, and ***winding*** $\mathbb{A}.w$. $\mathbb{A}.\vec{\mathbf{w}}$ returns the wector of $\mathbb{A}$. $\mathbb{A}.\mathbf{q}$ returns $\mathbb{A}.\mathbf{p} + \mathbb{A}.\vec{\mathbf{w}}$, the endpoint of the arrow. We say that $\mathbb{A}$ is a ***zero-turn*** arrow iff $\mathbb{A}.w$ is in $(-\pi, +\pi]$. We use the concatenation of two points without the overhead arrow to denote the zero-turn arrow between them. For example, $\mathbf{ab}$ denotes the zero-turn arrow, $\mathbb{A}$, with $\mathbb{A}.\mathbf{p} = \mathbf{a}$ and $\mathbb{A}.\mathbf{q} = \mathbf{b}$. We show $\mathbb{A}$ as an arrow from $\mathbb{A}.\mathbf{p}$ to $\mathbb{A}.\mathbf{q}$ or, in its ***extended*** form, $e(\mathbb{A})$, from $\mathbb{A}.\mathbf{p} - \mathbb{A}.\vec{\mathbf{w}}$ to $\mathbb{A}.\mathbf{q}$, so that $\mathbb{A}.\mathbf{p}$ is the midpoint of $e(\mathbb{A})$. We show $\mathbb{A}.\mathbf{w}$ as a spiral around $\mathbb{A}.\mathbf{p}$.

2.2. Transformations

Planar ***transformations*** are written in curly caps ($\mathcal{U}$). Application of $\mathcal{U}$ to point $\mathbf{p}$ (resp. vector $\vec{\mathbf{v}}$) is written $\mathcal{U} \cdot \mathbf{p}$ (resp. $\mathcal{U} \cdot \vec{\mathbf{v}}$). Notation $\mathcal{U} \cdot \mathcal{V} \cdot \mathcal{W} \cdot \mathbf{p}$ stands for $\mathcal{U} \cdot (\mathcal{V} \cdot (\mathcal{W} \cdot \mathbf{p}))$, where composition, '$\cdot$', is not always commutative. $\mathcal{U}^t$ is the ***power*** of $\mathcal{U}$. $\mathcal{U}^0$ is the ***identity*** $\mathcal{I}$.

Let $\mathcal{T}$ be an affine-transformation. When $\mathcal{T}$ is a pure translation by $\vec{\mathbf{v}}$, $\mathcal{T} \cdot \mathbf{p} = \mathbf{p} + \vec{\mathbf{v}}$. Otherwise, there exists a ***fixed-point***, $\mathbf{f}$, such that $\mathcal{T} \cdot \mathbf{p} = \mathbf{f} + \mathcal{L}_{\mathcal{T}} \cdot \overrightarrow{\mathbf{fp}}$, where $\mathcal{L}_{\mathcal{T}}$ is referred to as the ***linear-part*** of $\mathcal{T}$.

The terms 'rotation', 'dilation', and 'similarity' (defined below) refer to special cases of affine-transformations in the plane, which may be used to transform points and vectors. The term 'twirl' refers to a generalization of a similarity. It operates on points, wectors, and arrows. When preceded with "linear-", these terms refer to their linear-parts.

Rotation by angle α about fixed-point $\mathbf{f}$ is written $\mathcal{R}_{\mathbf{f},\alpha}$ and may be constructed using $\mathrm{R}(\mathbf{f}, \alpha)$.

$$\mathcal{R}_{\mathbf{f},\alpha} \cdot \vec{\mathbf{v}} = \mathcal{R}_{\alpha} \cdot \vec{\mathbf{v}} \tag{1}$$

$$\mathcal{R}_{\mathbf{f},\alpha} \cdot \mathbf{p} = \mathbf{f} + \mathcal{R}_{\alpha} \cdot \overrightarrow{\mathbf{fp}} \tag{2}$$

where $\mathcal{R}_{\alpha}$ is the linear-rotation by α around the origin, and, hence, is the linear-part of $\mathrm{R}_{\mathbf{f},\alpha}$. Using shortcut $\mathcal{R}$ for $\mathcal{R}_{\frac{\pi}{2}}$, $\mathcal{R}_{\alpha} \cdot \vec{\mathbf{v}} = \cos\alpha\,\vec{\mathbf{v}} + \sin\alpha\,\mathcal{R} \cdot \vec{\mathbf{v}}$.

Dilation about the fixed point $\mathbf{f}$, by scaling ratio λ is written $\mathcal{D}_{\mathbf{f},\lambda}$ and constructed using $\mathrm{D}(\mathbf{f}, \lambda)$. $\mathcal{D}_{\mathbf{f},\lambda} \cdot \vec{\mathbf{v}} = \lambda\vec{\mathbf{v}}$.

$$\mathcal{D}_{\mathbf{f},\lambda} \cdot \mathbf{p} = \mathbf{f} + \lambda\overrightarrow{\mathbf{fp}} \tag{3}$$

Similarity about the fixed point $\mathbf{f}$, by ratio λ and by angle α is written $\mathcal{S}_{\mathbf{f},\lambda,\alpha}$ and constructed using $\mathrm{S}(\mathbf{f}, \lambda, \alpha)$.

$$\mathcal{S}_{\mathbf{f},\lambda,\alpha} = \mathcal{D}_{\mathbf{f},\lambda} \cdot \mathcal{R}_{\mathbf{f},\alpha} = \mathcal{R}_{\mathbf{f},\alpha} \cdot \mathcal{D}_{\mathbf{f},\lambda} \tag{4}$$

Twirl about the fixed point $\mathbf{f}$, by positive ratio λ, and winding α is written $\mathcal{T}_{\mathbf{f},\lambda,\alpha}$ and constructed using $\mathrm{T}(\mathbf{f}, \lambda, \alpha)$.

$$\mathcal{T}_{\mathbf{f},\lambda,\alpha} \cdot \mathbf{p} = \mathcal{S}_{\mathbf{f},\lambda,\alpha} \cdot \mathbf{p} \tag{5}$$

$$\mathcal{T}_{\mathbf{f},\lambda,\alpha} \cdot \vec{\mathbf{v}} = \mathcal{S}_{\mathbf{f},\lambda,\alpha} \cdot \vec{\mathbf{v}} \tag{6}$$

$$\mathcal{T}_{\mathbf{f},\lambda,\alpha} \cdot \mathbb{A}_{\mathbf{p},m,w} = \mathrm{A}(\mathcal{S}_{\mathbf{f},\lambda,\alpha} \cdot \mathbf{p}, \lambda m, \alpha + w) \tag{7}$$

$$\mathcal{S}^{t}_{\mathbf{f},m,w} = \mathcal{S}_{\mathbf{f},m^t,tw}. \tag{8}$$

Although similarity $\mathcal{S}$ may be represented by its ***fixed-point*** $\mathbf{f}$ and by its linear-part $\mathcal{L}$, it is often represented in ***homogeneous*** form by a 3×3 matrix, $H_{\mathcal{S}}$, composed of the 2×2 matrix-form of $\mathcal{L}$, by the column of the 2 coordinates of $\mathbf{f} + \mathcal{L} \cdot \overrightarrow{\mathbf{fg}}$, where $\mathbf{g}$ is the ***global origin***, and by $(0,0,1)$ as the bottom-row.

One could represent twirl $\mathcal{T}$ by the image $\mathbb{A} = \mathcal{T} \cdot \mathbb{I}$ of the ***identity arrow***, $\mathbb{I} = \mathrm{A}(\mathbf{g}, \mathrm{P}(1,0))$. Instead, for clarity, we represent twirl $\mathcal{T}_{\mathbf{f},\lambda,\alpha}$ explicitly by $\mathbf{f}$, m, and w. When $\mathcal{T}$ is a zero-turn twirl, its m and w may be easily derived from the matrix form of its linear-part, $\mathcal{L}_{\mathcal{T}}$.

2.3. Lagrange interpolation of three timed constraints

The timed ***Lagrange interpolation*** [1], $v(t)$, of 3 reals (so that $v(a) = v_a$, $v(b) = v_b$, and $v(c) = v_c$) is a quadratic function of time:

$$v(a, v_a, b, v_b, c, v_z, t) = xv_a + yv_b + zv_z, \tag{9}$$

with $x + y + z = 1$ and

$$x = \frac{t-b}{a-b}\frac{t-c}{a-c}, y = \frac{t-a}{b-a}\frac{t-c}{b-c}, z = \frac{t-a}{c-a}\frac{t-b}{c-b} \tag{10}$$

Applying it to the corresponding Cartesian coordinates of points, one obtains a ***Parabolic Point-Motion***, $\mathbf{p}(t)$, that interpolates three timed keyframes, $\mathbf{p}(a) = \mathbf{p}_a$, $\mathbf{p}(b) = \mathbf{p}_b$, and $\mathbf{p}(c) = \mathbf{p}_c$. The moving point is the ***Weighted Arithmetic Mean*** (***WAM***) of the keyframe points:

$$\mathbf{p}(a, \mathbf{p}_a, b, \mathbf{p}_b, c, \mathbf{p}_c, t) = x\mathbf{p}_a + y\mathbf{p}_b + z\mathbf{p}_c \tag{11}$$

2.4. Neville reformulation

The above Parabolic Point-Motion may also be defined using the equivalent ***Neville*** construction as a composition of 3 ***timed linear morphs*** between points:

$$p(t) = \mathrm{L}(\mathrm{L}(\mathbf{p}_a, \mathbf{p}_b, q), \mathrm{L}(\mathbf{p}_b, \mathbf{p}_c, r), s), \tag{12}$$

where L is the ***LERP***

$$\mathrm{L}(\mathbf{a}, \mathbf{b}, t) = \mathbf{a} + t\overrightarrow{\mathbf{ab}} \tag{13}$$

and

$$q = \frac{t-a}{b-a}, r = \frac{t-b}{c-b}, s = \frac{t-a}{c-a} \tag{14}$$

2.5. Steady Morph (SMA) and Steady Average of 2 Arrows

Steadiness for animated twirls, defined below, is central to several of the solutions proposed in this paper. The general concept of ***steadiness*** has been defined previously for affinity-motions that morph between two affinities [2] or equivalently between two triangles, for planar maps [3], for non-planar maps [4], and for lattices [5]. It has also been suggested, as a suitable goodness-measure, for optimizing the overall motions of dancers in group choreographies [6].

Consider two arrows, $\mathbb{A}$ and $\mathbb{B}$ and two ***knots*** (time-values), a and b. Assume that we want to define a smooth motion, i.e., time-parameterized arrow, $\mathbb{D}(t)$ that satisfies 2 keyframe constraints: $\mathbb{D}(a) = \mathbb{A}$ and $\mathbb{D}(b) = \mathbb{B}$.

In cases where $\mathbb{A}.\vec{\mathbf{w}} = \mathbb{B}.\vec{\mathbf{w}}$, we have a pure translation and (using q from Eq. 14 and arrow constructor from Sec. 2.1):

$$\mathbb{D}(t) = \mathrm{A}(\ (1-q)\mathbb{A}.\mathbf{p} + q\mathbb{B}.\mathbf{p}\ ,\ \mathbb{A}.m\ ,\ \mathbb{A}.w\ ) \tag{15}$$

Otherwise, i.e., in non-singular cases where $\mathbb{A}.\vec{\mathbf{w}} \neq \mathbb{B}.\vec{\mathbf{w}}$, we use a ***steady twirl*** (with q from Eq. 14):

$$\mathbb{D}(t) = \mathcal{T}^q_{\mathbb{A},\mathbb{B}} \cdot \mathbb{A} \tag{16}$$

where $\mathcal{T}_{\mathbb{A},\mathbb{B}}$ is the twirl that maps $\mathbb{A}$ to $\mathbb{B}$ and may be computed [3] as $\mathrm{T}(\mathbf{f}, m, w)$ with:

$$m = \frac{\mathbb{B}.m}{\mathbb{A}.m}, \tag{17}$$

$$w = \mathbb{B}.w - \mathbb{A}.w, \tag{18}$$

$$\mathbf{f} = \mathbf{a} + \frac{\mathrm{V}(\vec{\mathbf{w}} \bullet \vec{\mathbf{u}}, (\mathcal{R} \cdot \vec{\mathbf{w}}) \bullet \vec{\mathbf{u}})}{d}, \tag{19}$$

with $\vec{\mathbf{u}} = \mathbb{B}.\mathbf{p} - \mathbb{A}.\mathbf{p}$, $\vec{\mathbf{w}} = \langle m\cos w - 1, m \sin w\rangle$, and $d = \vec{\mathbf{w}}^2$.

We define, below two arrow-valued functions: one that morphs between two arrows and one that computes the steady similarity of such a motion and applies it to a third arrow, so as to "transport" it. We will use these functions to formulate some of our PITA solutions.

$$\mathrm{M}(\mathbb{A}, \mathbb{B}, t) = \mathcal{T}^t_{\mathbb{A},\mathbb{B}} \cdot \mathbb{A}, \tag{20}$$

is the ***Steady Morph between two Arrows*** (***SMA***), shown in Fig. 3, where $\mathbb{D}(0) = \mathbb{A}$ and $\mathbb{D}(1) = \mathbb{B}$.

$$\mathrm{M}(a, \mathbb{A}, b, \mathbb{B}, t) = \mathrm{M}(\mathbb{A}, \mathbb{B}, q) = \mathcal{T}^q_{\mathbb{A},\mathbb{B}} \cdot \mathbb{A}, \tag{21}$$

using q from Eq. 14, is the ***timed-SMA***, where $\mathbb{D}(a) = \mathbb{A}$ and $\mathbb{D}(b) = \mathbb{B}$. Note that SMA is symmetric and hence, for all t,

$$\mathrm{M}(a, \mathbb{A}, b, \mathbb{B}, t) = \mathrm{M}(b, \mathbb{B}, a, \mathbb{A}, t). \tag{22}$$

We said above that, to morph between $\mathbb{A}$ and $\mathbb{B}$, we use a translation when $\mathbb{A}.\vec{\mathbf{w}} == \mathbb{B}.\vec{\mathbf{w}}$ and an SMA otherwise. This is correct. But, in practice, one needs to identify (and treat using a translation) cases where $\mathbb{A}.\vec{\mathbf{v}}$ and $\mathbb{B}.\vec{\mathbf{v}}$ are almost identical, because in such almost-singular configurations, the computation of the fixed-point $\mathbf{f}$ may be numerically unstable.

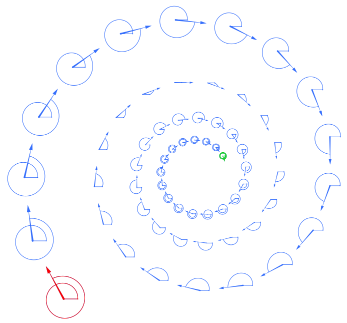


Figure 3: Intermediate frames (blue) of a ***Steady Morph between two Arrows*** (***SMA***), $\mathrm{M}(\mathbb{A}, \mathbb{B}, t)$, from the red arrow to the short green one. Observe that linear evolution of the winding (shown as a coil) and the steadiness of the pattern: Coordinate systems defined by consecutive frames are related by a constant similarity. The velocity of any moving point remains constant in the moving frame. The origin of the arrow (and in fact any moving point) traces a logSpiral.

We define the ***Steady Average*** of $\mathbb{A}$ and $\mathbb{B}$ as

$$\mathrm{M}(\mathbb{A}, \mathbb{B}) = \mathrm{M}(\mathbb{A}, \mathbb{B}, 1/2) \tag{23}$$

2.6. Steady Transport (STA) of an Arrow

The ***Steady Transport of an Arrow*** (***STA***), which moves arrow $\mathbb{C}$ by the similarity $\mathrm{M}(\mathbb{A}, \mathbb{B}, t)$, is defined by:

$$\mathrm{T}(\mathbb{C}, \mathbb{A}, \mathbb{B}, t) = \mathcal{T}^t_{\mathbb{A},\mathbb{B}} \cdot \mathbb{C}. \tag{24}$$

The ***timed-STA*** is (using q from Eq: 14):

$$\mathrm{T}(\mathbb{C}, a, \mathbb{A}, b, \mathbb{B}, t) = \mathrm{T}(\mathbb{C}, \mathbb{A}, \mathbb{B}, q) = \mathcal{T}^q_{\mathbb{A},\mathbb{B}} \cdot \mathbb{C}. \tag{25}$$

2.7. Non-steady variant of SMA

Given a fixed point, $\mathbf{f}$, of the twirl between $\mathbb{A}$ and $\mathbb{B}$, the SMA (Eq. 20) involves two morphs between different pairs of wectors: the pair ($\mathbb{A}.\vec{\mathbf{w}}$,$\mathbb{B}.\vec{\mathbf{w}}$), and the pair ($\mathbb{A}.\mathbf{p}-\mathbf{f}$,$\mathbb{B}.\mathbf{p}-\mathbf{f}$). In both cases, SMA interpolates their magnitudes using the ***weighted geometric mean*** (***WGM***), which is equivalent to using a weighted arithmetic mean of their logarithms. In Fig. 4, we compare this steady solution two non-steady variants: The ***weighted arithmetic mean*** (***WAM***) and the ***weighted harmonic mean*** (***WHM***). Both modify the bulge of the curve and the spacing of the frames near the shorter (green) arrow.

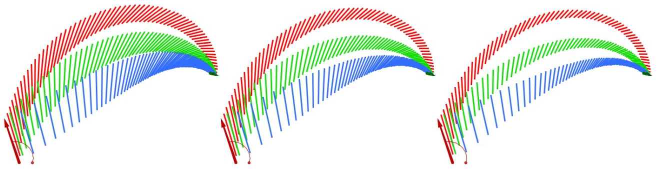


Figure 4: We show 9 morphs between red (left) and green (right) arrows using the Weighted (Pythagorean) Arithmetic (WAM), Geometric (WGM), or Harmonic (WHM) Means for the path of the center of the arrow: WAM (left), WGM (center), WHM (right), and for the magnitude of its wector: WAM (red), WGM (greed), WHM (blue).

2.8. Exponential mapping for geometric averaging

The WGM, $a^x b^y c^z$, of values a, b, and c with weights x, y, and z (defined in Eq. 10) may be computed using the WAM of logs as $e^{x\log(a)+y\log(b)+z\log(c)}$. Some prior art (see 5) builds on this idea, but instead of reals (a, b, and c) use square matrices, which, for keyframe arrow $\mathbb{A}$ either represent the linear part, $L(\mathbb{A})$, of the transformations associated with $\mathbb{A}$ or the homogeneous form, $H(\mathbb{A})$ of that transformation. The computation of the logarithm and exponential of these matrices is discussed in Appendix A.

3. Our basket of 8 chosen PITAs

We have chosen to compare the following 8 PITAs:

- P1:Pa-Ca $\mathbf{p}$=Parabolic, $\vec{\mathbf{w}}$=Cartesian
- P2:LS-Po $\mathbf{p}$=LogSpiral, $\vec{\mathbf{w}}$=Polar
- P3:Pa-LP $\mathbf{p}$=Parabolic, $\vec{\mathbf{w}}$=LogPolar
- P4:Be-LP $\mathbf{p}$=Bezier, $\vec{\mathbf{w}}$=LogPolar
- P5:He-LP $\mathbf{p}$=Homogeneous exponential, $\vec{\mathbf{w}}$=LogPolar
- P6:La-LP $\mathbf{p}$=Lagrange, $\vec{\mathbf{w}}$=LogPolar
- P7:Ra-Pe $\mathbf{p}$=Radial-perseverant, $\vec{\mathbf{w}}$=Perseverant
- P8:Lo-Pe $\mathbf{p}$=Local-perseverant, $\vec{\mathbf{w}}$=Perseverant

We use 4-letter mnemonics to identify the animation of the start-point $\mathbf{p}$ (first 2 letters: Pa-, LS-, Be-, Ne-, La-, Ra-, Lo-) and of the wector $\vec{\mathbf{w}}$ (last 2 letters: -Ca, -Po, -LP, -Pe). (These 4 wector solutions are shown in Fig. 5.

We first discuss 3 ***independent*** PITAs, P123, in which these solutions are fully independent. Then, we discuss 3 ***coordinated*** PITAs, P456, which have the same wector animation, but different motions of $\mathbf{p}$. Finally, we discuss 2 ***perseverant*** PITAs, P78, in which the acceleration of the moving wector is constant in the moving frame.

For each PITA, Px, we propose a formulation of the motion (time-varying arrow), $\mathbb{D}(t) = \text{Px}(a, \mathbb{A}, b, \mathbb{B}, c, \mathbb{C}, t)$, such that $\mathbb{D}(a) = \mathbb{A}$, $\mathbb{D}(b) = \mathbb{B}$, and $\mathbb{D}(c) = \mathbb{C}$ (Fig. 1 compares the 8 motions for the same keyframes with uniform timing).

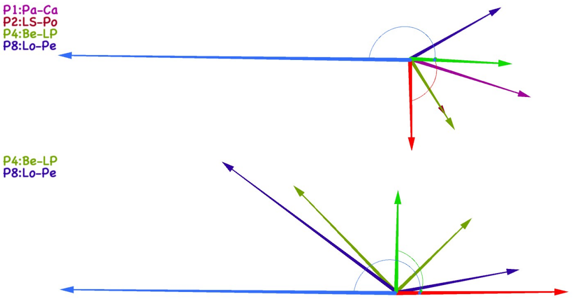


Figure 5: Top: $\mathbb{D}(1/4).\vec{\mathbf{w}}$ for -Ca, -Po, -LP, and -Pe: -Po and -LP are parallel. Bottom: $\mathbb{D}(1/4).\vec{\mathbf{w}}$ and $\mathbb{D}(3/4).\vec{\mathbf{w}}$ for -LP, and -Pe.

3.1. Independent PITAs: P123

We discuss here two options (Pa- and LS-) for animating $\mathbb{D}(t).\mathbf{p}$ and three options (-Ca, -Po, and -LP) for animating $\mathbb{D}(t).\mathbf{w}$. From the 6 possible combinations of these, we chose three: P1:Pa-Ca, P2:LS-Po, and P3:Pa-LP.

Three-Point Interpolations:

For readability, let $\mathbf{p}_a = \mathbb{A}.\mathbf{p}$, $\mathbf{p}_b = \mathbb{B}.\mathbf{p}$, and $\mathbf{p}_c = \mathbb{C}.\mathbf{p}$. We want $= \mathbb{D}(t).\mathbf{p}$ to be a ***point-motion***, $\mathbf{p}(t)$, satisfying $\mathbf{p}(a) = \mathbf{p}_a$, $\mathbf{p}(b) = \mathbf{p}_b$, and $\mathbf{p}(c) = \mathbf{p}_c$.

Pa- uses a Parabolic Point-Motion (Eq. 11) for $\mathbf{p}(t)$, which thus traces a parabola and has constant acceleration in the global ***frame*** (We call it ***global-perseverance***).

LS- uses a ***LogSpiral Motion*** (***LSM***) for $\mathbf{p}(t)$ (Fig. 6). We compute it using

$$\mathbf{p}(t) = \mathcal{T}_{\mathbf{ab},\mathbf{bc}}^{\frac{t-a}{b-c}\left(\frac{t-c}{b-a}-2\frac{t-b}{c-a}\right)} \cdot \mathbf{a}. \tag{26}$$

where $\mathcal{T}_{\mathbf{ab},\mathbf{bc}}$ is an SMA (Eq. 21) that morphs steadily between zero-turn (Sec. 2.1) arrows $\mathbf{ab}$ and $\mathbf{bc}$. One can verify that this motion satisfies the 3 interpolation constraints.

Changing b amounts here to a linear time-warp and thus keeps the intermediate frames (red samples) on the unique LogSpiral curve through the 3 points. Hence, solutions based on LS- are not truly ***knot-controlled*** (Sec. 4.8).

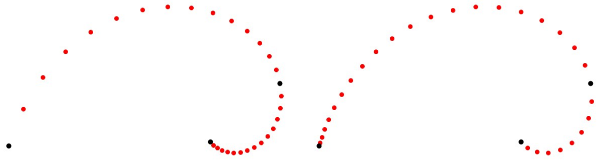


Figure 6: LogSpiral Motion through 3 timed-keyframes (black control-points) for $b = (a + c)/2$ (left) and $b = a + 0.7(c - a)$ (right).

Three-Wector Interpolations:

Let $\vec{\mathbf{w}}_a = \mathbb{A}.\vec{\mathbf{w}}$, $\vec{\mathbf{w}}_b = \mathbb{B}.\vec{\mathbf{w}}$, and $\vec{\mathbf{w}}_c = \mathbb{C}.\vec{\mathbf{w}}$. We want $= \mathbb{D}(t).\vec{\mathbf{w}}$ to be a ***wector-motion***, $\vec{\mathbf{w}}(t)$, satisfying $\vec{\mathbf{w}}(a) = \vec{\mathbf{w}}_a$, $\vec{\mathbf{w}}(b) = \vec{\mathbf{w}}_b$, and $\vec{\mathbf{w}}(c) = \vec{\mathbf{w}}_c$.

-Ca replaces wectors by corresponding zero-turn vectors. It uses a Lagrange interpolation (Eq. 9) for the Cartesian components of these. Hence, solutions based on -Ca are not ***winding-aware*** (Sec. 4.6).

-Po uses Lagrange interpolations (Eq. 9) to blend the polar coordinates (m, w) of $\vec{\mathbf{w}}_a$, $\vec{\mathbf{w}}_b$, and $\vec{\mathbf{w}}_c$.

-LP uses Lagrange interpolations to blend the logPolar coordinates ($\log m$, w) of $\vec{\mathbf{w}}_a$, $\vec{\mathbf{w}}_b$, and $\vec{\mathbf{w}}_c$.

Chosen independent PITAs:

P1:Pa-Ca uses Pa- for $\mathbb{D}(t).\mathbf{p}$ and -Ca for $\mathbb{D}(t).\vec{\mathbf{w}}$. Hence, $\mathbb{D}(t).\mathbf{q}$ follows its own Three-Point Interpolation.

P2:LS-Po uses LS- for $\mathbb{D}(t).\mathbf{p}$ and -Po for $\mathbb{D}(t).\vec{\mathbf{w}}$. Note that, because some users may prefer this visualization, in Fig. 1, we use a simple variant of P2, which uses LS- for the midpoint, $\mathbb{D}(t).\mathbf{p} + \frac{1}{2}\mathbb{D}(t).\vec{\mathbf{w}}$, of the arrow and we show the extended version of the keyframes and of the moving arrow, but draw the curve traced by $\mathbb{D}.\mathbf{p}$.

P3:Pa-LP uses Pa- for $\mathbb{D}(t).\mathbf{p}$ and -LP for $\mathbb{D}(t).\vec{\mathbf{w}}$.

These 3 PITAs are compared to other combinations of the point and wector interpolations presented above (Fig. 7).

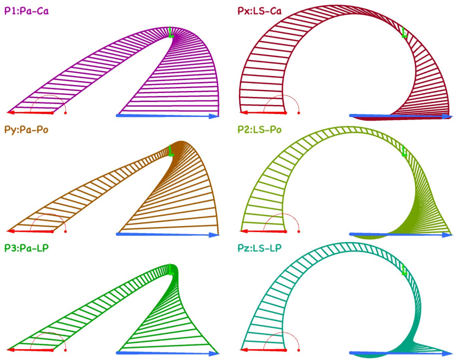


Figure 7: Independent combinations of animations for $\mathbb{D}(t).\mathbf{p}$: Pa- (left column) and LS- (right column) with animations for $\mathbb{D}(t).\vec{\mathbf{w}}$: -Ca (top), -Po (middle), and -LP (bottom).

3.2. Coordinated PITAs: P456

The coordinated PITAs discussed here use the -LP Three-Wector Interpolation. Hence, they have the same $\mathbb{D}(t).\vec{\mathbf{w}}$ as P3. The sometimes significant disparity of their $\mathbb{D}(t).\mathbf{p}$ motion is due to the lack of commutativity of the similarity transformations used to compute them.

P4:Be-LP is our adaptation (to similarities) of the ***BiSAM*** solution [2] P4 returns:

$$\text{Bezier}(\mathbb{A}, \mathbb{L}, \mathbb{B}, 2u)\,\text{if}(u \leq 1/2) \tag{27}$$

$$\text{Bezier}(\mathbb{B}, \mathbb{R}, \mathbb{C}, 2u-1)\,\text{otherwise} \tag{28}$$

with $u = (t-a)/(c-a)$,

$$\text{Bezier}(A, B, C, t) = \text{M}(\text{M}(A, B, t), \text{M}(B, C, t), t), \tag{29}$$

$$\mathbb{L} = \text{H}(\mathbb{A}, \mathbb{B}, \mathbb{C}), \tag{30}$$

$$\mathbb{R} = \text{H}(\mathbb{C}, \mathbb{B}, \mathbb{A}), \text{and} \tag{31}$$

$$\text{H}(\mathbb{A}, \mathbb{B}, \mathbb{C}) = \text{T}(\mathbb{C}, \mathbb{B}, \text{M}(\mathbb{A}, \mathbb{B}, 3/4), 1/4), \tag{32}$$

with M() defined in Eq. 20 and T() in Eq. 24.

P5:He-LP uses a WAM of the logs of the homogeneous matrices (Sec. 2.8), $H(\mathbb{A})$, $H(\mathbb{B})$, and $H(\mathbb{C})$, of the keyframes:

$$H(t) = e^{x \log H(\mathbb{A}) + y \log H(\mathbb{B}) + z \log H(\mathbb{C})}, \tag{33}$$

but with respect to a local origin, to make it similarity-invariant. The details are in Appendix C.

P6:La-LP is $\text{M}(\mathbb{L}, \mathbb{R})$, Steady Average defined in Eq. 23, of $\mathbb{L} = \text{T}_1(a, \mathbb{A}, b, \mathbb{B}, c, \mathbb{C}, t)$ and $\mathbb{R} = \text{T}_1(c, \mathbb{C}, b, \mathbb{B}, a, \mathbb{A}, t)$, where

$$\text{T}_1(a, \mathbb{A}, b, \mathbb{B}, c, \mathbb{C}, t) = \text{T}(\text{M}(\mathbb{A}, \mathbb{B}, y+z), \mathbb{B}, \mathbb{C}, z) \tag{34}$$

using $\text{T}(\mathbb{C}, \mathbb{A}, \mathbb{B}, t)$, the STA defined in Eq. 24, and ratios y and z defined in Eq. 10. We invented this formulation by arranging the Lagrange interpolant (Eq. 11) into point-plus-2-vector form ($x\mathbf{a} + y\mathbf{b} + z\mathbf{c} = b + x\overrightarrow{\mathbf{ba}} + z\overrightarrow{\mathbf{bc}}$) and mapping points to arrows and vectors to twirls. We compare it, in Appendix D, to variants corresponding to the 11 permutations of this form, such as $b + z\overrightarrow{\mathbf{bc}} + x\overrightarrow{\mathbf{ba}}$ or $b + (x+z)\overrightarrow{\mathbf{ba}} + z\overrightarrow{\mathbf{ac}}$, and to solutions derived from the Neville formulation. In (Fig. 8, we compare P3456, which are all -LP).

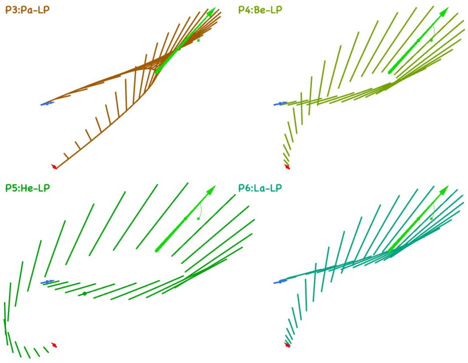


Figure 8: Comparing the four -LP PITAs: P3456.

3.3. Perseverant PITAs: P78

We say that a motion (time-parameterized similarity) is ***globally-perseverant*** (resp. ***radially-perseverant***, ***locally-perseverant***) iff the acceleration of a moving point, $\mathbf{p}(t)$, remains constant when expressed in the global- (resp. radial-, local)- frame. By ***local-frame***, we mean the moving coordinate system defined by $\mathbb{D}(t)$. By ***radial-frame***, we mean the moving coordinate system defined by arrow $\mathbf{fp}(t)$, for some fixed-point $\mathbf{f}$. P1 is globally-perseverant. P7 is radially-perseverant. P8 is locally-perseverant. P78 have the same vector motion, $\mathbb{D}(t).\vec{\mathbf{w}}$, which is locally-perseverant (-Pe): The second derivative of $\mathbb{D}(t).\vec{\mathbf{w}}$ is constant in the local frame defined by $\text{A}(\mathbf{g}, \mathbb{D}(t).\vec{\mathbf{w}})$, where $\mathbf{g}$ is the global origin. Since they ignore the winding, we use "vector", rather than "wector", when discussing them.

P7:Ra-Pe is defined by ODE

$$\frac{d^2}{dt^2}\mathbf{p}(t) = \mathcal{S} \cdot \overrightarrow{\mathbf{fp}}(t). \tag{35}$$

where $\mathcal{S}$ is a constant linear-similarity and $\mathbf{f}$ a fixed-point at which the acceleration remains zero. $\mathcal{S}$ and $\mathbf{f}$ define a mapping from each point $\mathbf{p}$ onto a vector $\mathcal{S} \cdot \overrightarrow{\mathbf{fp}}$, so that, when a point moving under P7 passes through $\mathbf{p}$, it experiences an acceleration given by that vector.

The general form of the solution of Eq. 35 is:

$$\mathbf{p}(t) = \mathbf{f} + \mathcal{S}_1(t) \cdot \overrightarrow{\mathbf{fp}}(0) + \mathcal{S}_2(t) \cdot \mathbf{p}'(0), \tag{36}$$

where $\mathbf{p}'(0)$ represents the initial velocity vector of point $\mathbf{p}$. The 2×2 matrices $\mathcal{S}_1(t)$ and $\mathcal{S}_2(t)$ vary with time t.

In Appendix E, we provide the derivation of the closed-form expressions for computing $\mathcal{S}_1(t)$, and $\mathcal{S}_2(t)$, $\mathbf{f}$, and $\mathbf{p}'(0)$, so as to satisfy the three interpolation constraints. In Fig. 9, we compare P78.

P8:Lo-Pe uses the ***Persistent Affine Motion*** (***PAM***) described and called "UAM" in Muthumanickam's Master's Thesis [7]. PAM interpolates 3 affinities, $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, at uniformly-spaced knots, satisfying $\mathcal{U}(0) = \mathcal{A}$, $\mathcal{U}(\frac{1}{2}) = \mathcal{B}$, and $\mathcal{U}(1) = \mathcal{C}$. When affinities $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are represented each by a triangle, PAM defines the motion of a triangle.

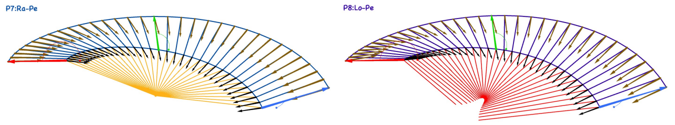


Figure 9: Comparing P7 (left), which is radially-perseverant, and P8 (right), which is locally-perseverant, We show their acceleration vectors at $\mathbb{D}(t).\mathbf{p}$ (black) and at $\mathbb{D}(t).\mathbf{q}$ (brown). We shown (left) orange line-segments from $\mathbb{D}(t).\mathbf{p}$ to the global fixed point of P7. Observe the constant relation (radial perseverance) between each black arrow and the corresponding orange segment. We shown (right) red line-segments from $\mathbb{D}(t).\mathbf{p}$ to the instantaneous (local) fixed point of P8. Observe the constant relation (local perseverance) between each black arrow and the corresponding red segment.

The PAM motion $\mathbf{p}(t)$ of an arbitrary starting point $\mathbf{p}_0$ in homogeneous coordinates is defined by

$$\mathbf{p}(t) = \mathcal{U}(t) \cdot \mathbf{p}_0, \tag{37}$$

where $\mathcal{U}(t)$ is a time-varying affinity. Its instantaneous acceleration is

$$\frac{d^2}{dt^2}\mathbf{p}(t) = \mathcal{U}''(t) \cdot \mathbf{p}_0. \tag{38}$$

Imposing that the acceleration be constant in the moving frame can then be written as:

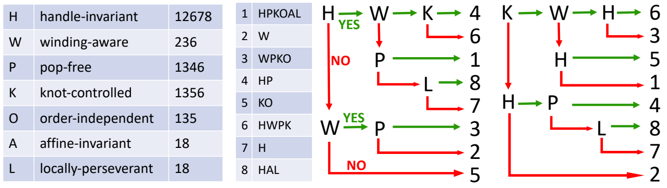
$$\mathcal{U}''(t) = \mathcal{U}(t) \cdot \mathcal{E} \tag{39}$$

where $\mathcal{E}$ is a constant affinity (which we represent by a homogeneous matrix) that is independent of $\mathbf{p}_0$. This is a linear ODE of second order, that we solve in the Appendix F, where we give closed-form expressions of the entries in $\mathcal{U}(t)$ and $\mathcal{E}$ that ensure that the interpolation conditions $\mathcal{U}(0) = \mathcal{A}$, $\mathcal{U}(\frac{1}{2}) = \mathcal{B}$ and $\mathcal{U}(1) = \mathcal{C}$ are met. A PAM solution may not exist for some extreme configurations where the triangles have widely different orientations and shapes.

For P8, we require that all three key-triangles be similar and define P8 as the motion of a zero-turn arrow. Note that the position of the tip of each keyframe triangle is not important, as long as all 3 keyframe triangles are similar. This P8 restriction of PAM to similarities always exists.

4. Mathematical properties of the chosen PITAs

In this section, we discuss 11 mathematical properties that may be relevant to some applications. Seven of these are not satisfied by one or more PITAs (Fig. 10).



H	handle-invariant	12678
W	winding-aware	236
P	pop-free	1346
K	knot-controlled	1356
O	order-independent	135
A	affine-invariant	18
L	locally-perseverant	18

1	HPKOAL
2	W
3	WPKO
4	HP
5	KO
6	HWPK
7	H
8	HAL

Figure 10: PITAs per property, the inverse, and two decision-trees.

4.1. Always exists

All chosen PITAs ***exist*** (i.e., are uniquely defined and may be computed using closed-form expressions) for all time-values and for all configurations of keyframes (except possibly for some isolated singular cases). As discussed in Sec. 2.5, in the implementation of PITAs that rely on the Steady Morph between Arrows, to avoid numerical instability, we detect near-singular configurations and perform the morph using a linear morph.

4.2. Time-continuous and infinitely differentiable

A PITA is ***time-continuous*** iff for any choice of keyframes, the resulting motion is a continuous function of t. Time-continuity is important because it means that the animation is continuous (no jump). All our PITAs are time-continuous. In fact, they are infinitely differentiable.

4.3. Similarity-invariant

Px is ***similarity-invariant*** if, for any similarity $\mathcal{S}$,

$$\text{Px}(a, \mathcal{S} \cdot \mathbb{A}, b, \mathcal{S} \cdot \mathbb{B}, c, \mathcal{S} \cdot \mathbb{C}, t) = \mathcal{S} \cdot \text{Px}(a, \mathbb{A}, b, \mathbb{B}, c, \mathbb{C}, t) \tag{40}$$

All chosen PITAs are similarity-invariant because all their construction steps are. Similarity invarience is important because the designer should not have to worry about the global coordinate system and should be able to translate, rotate, or dilate the set of keyframes at will and to expect that the resulting animation or pattern (of arrows or shapes) will be transformed accordingly.

4.4. Symmetric

A PITA, P, is ***symmetric*** iff it is not altered by swapping $\mathbb{A}$ and $\mathbb{C}$ and replacing t (resp. a b, c), by $u(t)$ (resp. $u(a)$ $u(b)$, $u(c)$), with $u(t) = a + c - t$, i.e., iff

$$\text{P}(a, \mathbb{A}, b, \mathbb{B}, c, \mathbb{C}, t) = \text{P}(a, \mathbb{C}, c{+}a{-}b, \mathbb{B}, c, \mathbb{A}, c{+}a{-}t). \tag{41}$$

Lack of symmetry produces surprising, possibly frustrating, results for symmetric configurations of keyframes. All chosen PITAs are symmetric. (P6 was obtained by averaging two carefully matched asymmetric solutions.) Many candidates discussed in Appendix C and Appendix D are not symmetric.

4.5. Handle-invariant

Assume that we have 3 similar key-shapes, $\mathbf{S}_\text{a}$, $\mathbf{S}_\text{b}$, and $\mathbf{S}_\text{c}$. We want to use a PITA to define an animated shape $\mathbf{S}(t)$ such that $\mathbf{S}(a) = \mathbf{S}_\text{a}$, $\mathbf{S}(b) = \mathbf{S}_\text{b}$, and $\mathbf{S}(c) = \mathbf{S}_\text{c}$. To do so, we place a ***handle*** (keyframe $\mathbb{A}$) between two arbitrarily chosen points of $\mathbf{S}_\text{a}$. We place $\mathbb{B}$ between the two corresponding points of $\mathbf{S}_\text{b}$ and place $\mathbb{C}$ between the two corresponding points of $\mathbf{S}_\text{c}$. For any time t, we evaluate $\mathbb{D}(t) = \text{Px}(a, \mathbb{A}, b, \mathbb{B}, c, \mathbb{C}, t)$, and transform $\mathbf{S}_\text{a}$ by the twirl $\mathcal{T}_{\mathbb{A},\mathbb{D}(t)}$ between $\mathbb{A}$ and $\mathbb{D}(t)$. A PITA is ***handle-invariant*** iff $\mathbf{S}(t)$ is independent of the placing of the ***handle*** (the two points chosen to define $\mathbb{A}$) with respect to $\mathbf{S}_\text{a}$. All chosen PITAs are handle-invariant, except for P235 (Fig. 11).

4.6. Winding-aware

A PITA is ***winding-aware*** if it takes into account the windings (i.e., the turn-counts) of the keyframes and hence allows the designer to create motions during which $\mathbb{D}(t).w$ makes turns of more than 2π. P236 are winding-aware.

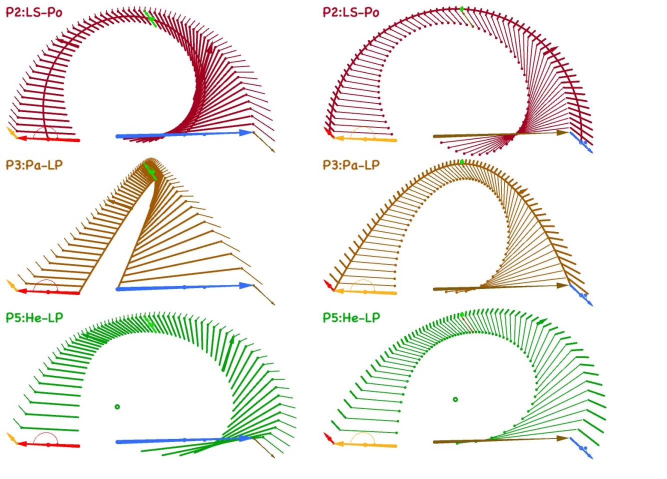


Figure 11: Swapping the red (left column) and the orange (right) handle-arrows shows that P2 (top row), P3 (middle), and P5 (bottom) are not handle-invariant.

4.7. Pop-free

A PITA is ***pop-free*** (control-continuous) if, for all t, $\mathbb{D}(t)$ is a continuous functions of all parameters of all keyframes. Control-continuity is essential for interactive editing and also for ensuring the time-continuity of animations of PITA-patterns that are controlled by 3 keyframes, which each are animated (for example by different PITAs). All chosen PITAs are pop-free, except for P278, for which changes in trigonometric branching produce abrupt changes in the pattern of PITA frames for some minute changes in the position, orientation, or magnitude of a keyframe (Fig. 12).

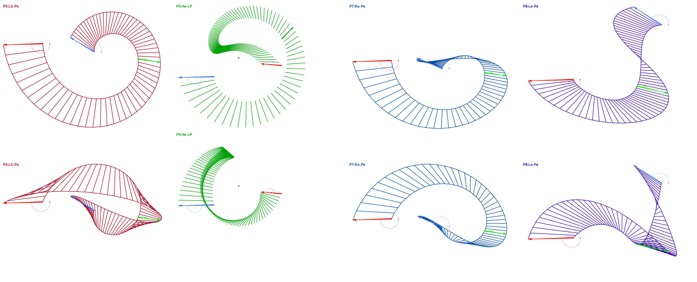


Figure 12: Tweaking the configuration (top) slightly (bottom) produces a pop for P2 (left), P5 (center-left), P7 (center-right) and P8 (right).

4.8. Knot-controlled

A PITA is ***knot-controlled*** if changing ratio $\frac{b-a}{c-a}$ changes the set of frames, $\mathbb{D}(t)$, that it produces. P1356 are knot-controlled. For P2, changes to that ratio affect $\mathbb{D}(t).\mathbf{p}$, but do not affect the logSpiral curve upon which $\mathbb{D}(t).\mathbf{p}$ travels (Fig. 13).

4.9. Order-independent

A PITA is ***order-independent*** if it is symmetric and if the animation it produces is also not affected by swapping $(a, \mathbb{A})$ and $(b, \mathbb{B})$, i.e., iff

$$\mathrm{P}(b, \mathbb{B}, a, \mathbb{A}, c, \mathbb{C}, t) = \mathrm{P}(a, \mathbb{A}, b, \mathbb{B}, c, \mathbb{C}, t). \tag{42}$$

Order-independence may be useful in applications in which we do not know in advance the order of the knots a, b and c. Only P135 are order-independent.

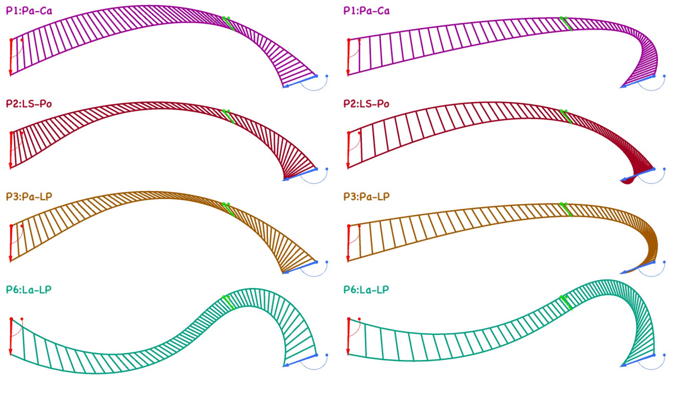


Figure 13: We show (top-down) P1236 for $\frac{b-a}{c-a} = 0.66$ (left) and $\frac{b-a}{c-a} = 0.40$ (right). The (top) curve traced by $\mathbb{D}(t).\mathbf{p}$ of P2 is not affected by this knot-change. Hence, only P1356 are fully knot-controlled.

4.10. Affine-invariant

A PITA, Px, is ***affine-invariant*** iff, for any affinity $\mathcal{U}$,

$$\begin{aligned}&\mathrm{Px}(a, \mathcal{U} \cdot \mathbb{A}, b, \mathcal{U} \cdot \mathbb{B}, c, \mathcal{U} \cdot \mathbb{C}, t)\\ &\qquad = \mathcal{U} \cdot \mathrm{Px}(a, \mathbb{A}, b, \mathbb{B}, c, \mathbb{C}, t)\end{aligned} \tag{43}$$

Affine-invariance may be useful in applications where the designer may wish to stretch a PITA pattern in some direction. Only P18 are affine-invariant, since only these are defined in terms of affine-invariant operators. Observe that, to achieve affine0invariance for P8, one must apply $\mathcal{U}$, not only to the keyframes, but also to the tip points of their triangles.

4.11. Local perseverance

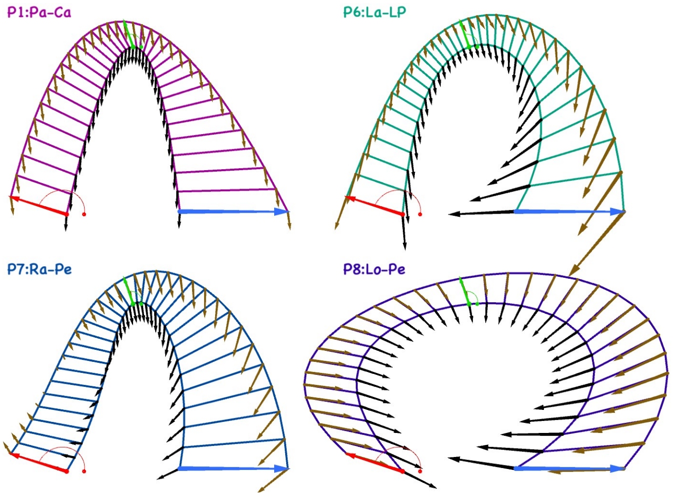


Figure 14: We show acceleration vectors at $\mathbb{D}(t).\mathbf{p}$ (black) and at $\mathbb{D}(t).\mathbf{q}$ (brown) for P1678.

In Sec. 3.3, we defined global-, radial-, and local-preseverance. P1 is globally-perseverant (constant acceleration). P7 is radially-perseverant. P8 is locally-perseverant. In Fig. 14, we compare the local perseverance of P8 to the radial perseverance of P7, to the global perseverance of P1, and to P6, which has none of these properties. Fig. 15 shows more clearly the difference between P7 and P8. P7 converges to a globally-perseverant motion when $\mathbf{f}$ goes to infinity and to a locally-perseverant motion when $\mathbb{A}.\mathbf{p} = \mathbb{B}.\mathbf{p} = \mathbb{C}.\mathbf{p}$.

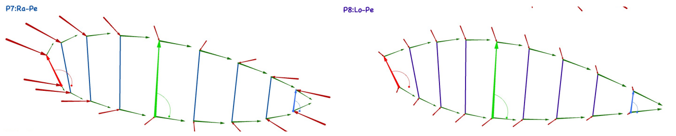


Figure 15: We compare P7 (left) and P8 (right) showing the velocities of $\mathbb{D}(t).\mathbf{p}$ and $\mathbb{D}(t).\mathbf{q}$ as dark-green vectors from them and their accelerations (pushing forces) as dark-red vectors arriving on them. Observe that, for P8, the shapes formed by $\mathbb{D}(t)$ and by the two accelerations are all similar to each other.

5. Relation to prior art

There is abundant literature on the interpolation of rotations and of rigid motions. Haarbach et. al [8] provide a clear and concise survey of these.

More infrequent is the analysis of the interpolation of similarities. An approach based on a WGM of the linear parts of similarity matrices of the keyframes was proposed in [9]. In Sec. Appendix B, we show that P10, which is based on a 2D version of the approach proposed in [9], produces results that are identical to P3 if the correct trigonometric branch is chosen. Our implementation of the logarithm and exponentiation operators used in P10 is based on a simpler derivation than that in [8]. We include the details of their computation in Appendix A.

However we are not aiming here to improve on these computations of interpolants, but rather to study and evaluate their results when applied to the design of curves or motions on the plane.

Below, for each chosen PITA, we give more specific references to relevant prior art.

P1:Pa-Ca is a 2D version of "component-wise linear blending rigid transformation matrices". The authors of [10] mention that this "most straightforward solution" has two drawbacks: it is non-rigid and may produce singular frames. P1 only interpolates the corresponding Cartesian components of the first and third vectors of the homogeneous matrix representation of each keyframe. The second vector is obtained trivially using a rotation by $\frac{\pi}{2}$. P1 is affine-invariant and handle-invariant.

P2:LS-Po is new as a combination. The LS- part was mentioned in [3]. The -Po part extends the scheme used in [11] for blending edge-lengths and vertex-angles from 2 to 3 keyframes and to a winding-aware interpolation.

P3:Pa-LP is related to prior solutions based on linear combinations of logs of the linear parts of matrices [2, 12]. The novelty here is that, since we operate in 2D and are restricted to similarities, we compute P3 using a simple formula that does not require computing logs and exponentials of matrices and that eliminates the branching problem, hence trivially making P3 winding-aware. It separates the interpolation of the linear part from the translation, as was done in [13].

P4:Be-LP is an adaptation to similarities of (what we call) the BiSAM approach proposed in [2]. The merit and novelty of this adaptation is that P4 extends BiSAM by making it winding-aware and hence also pop-free.

P5:He-LP is related to prior solutions that compute linear combinations of logarithms of homogeneous matrices [12, 9]. We provide the implementation details in Appendix A and explain our heuristics to make it similarity-invariant in Appendix C.

P6:La-LP is new.

P7:Ra-Pe is new.

P8:Lo-Pe is new. It is the only locally-perseverant solution.

6. Goodness and ease-of-use

In this section, we share suggestions for measuring the ease of using different PITAs for designing patterns and animations and the goodness of the results they produce.

6.1. Suitability for in-betweening automation

To assess whether the PITAs may be useful for the automation of **in-betweening** animation [14, 15], we consider the simplified scenario (Fig. 16), in which the lead-artist provides 3 timed-keyframes (represented by arrows $\mathbb{A} = \mathbb{D}(1)$, $\mathbb{B} = \mathbb{D}(3)$, and $\mathbb{C} = \mathbb{D}(5)$) of the position, orientation, and size of a shape. An in-betweening expert draws the two missing frames ($\mathbb{L}$ for $t = 2$ and $\mathbb{R}$ for $t = 4$). Then, for each PITA, we report an error measure formulated in terms of the **discrepancies** between $\mathbb{L}$ and $\mathbb{D}(2)$ and between $\mathbb{R}$ and $\mathbb{D}(4)$ and decide whether at least one of them provides an acceptable result. We conjecture that this naïve test may not be very useful, because, in non-trivial configurations, it is difficult, even for an expert, to know exactly where $\mathbb{L}$ and $\mathbb{R}$ should be and, more importantly, because the discrepancy metric may not correctly predict the quality of a finer interpolation. Hence, we may need to ask the expert to draw a whole pattern of intermediate frames and compare them to those produced by each one of our PITAs.

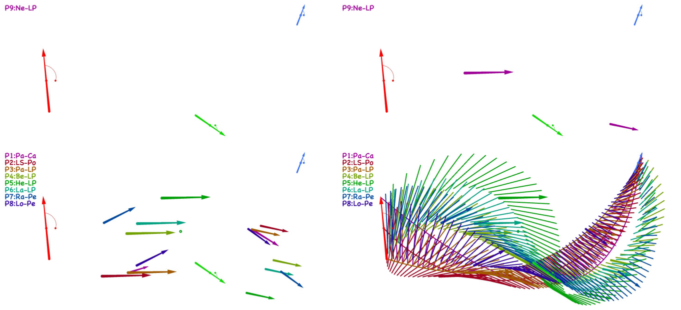


Figure 16: Given (top-left) the (red, green, blue) keyframes, the user guesses (top-right) the (brown, magenta) arrows $\mathbb{L}$ and $\mathbb{R}$. We show (bottom-left), for all PITAs, the actual arrows for $\mathbb{D}(2)$ and $\mathbb{D}(4)$ and, for context the other frames (bottom-right).

We have explored a variety of goodness measures for PITAs. We converged on measuring the discrepancy between each frame of a uniformly sampled PITA pattern and its ***prediction*** (when it exists), which may be: (1) the previous frame, (2) the steady similarity average between the previous and next frame, or (3) the prediction (Sec. 7.5) computed from the previous three frames. Measures based on options (1) and (2) are suitable for assessing the quality of non-steady motions. (Using them would be like using the length of an interpolating curve as a measure of its quality.) Measure (3) is excellent, but unfortunately unfair. Indeed, if we use the predictor for PITA Px, then Px will win since it will have zero discrepancy at each frame. Rating each PITA in terms of how well it does against predictors of all the other PITAs may seem democratic, but is unreliable, as it may be easily manipulated by dropping PITAs from the basket or adding to new PITAs or averages of existing ones.

6.2. Measure of goodness: Discrepancy

In the above discussions, we mentioned the need for measuring the discrepancy between two arrows. This is not a trivial task, especially because we want the measure to be similarity-invariant. Several formulae for computing the error between two coplanar edges are compared in [16]. We propose here a new (as far as we know) formulation of the ***discrepancy*** $\mathrm{E}(\mathbb{A},\mathbb{B})$ between two arrows (oriented line-segments):

$$\mathrm{x}(\mathbf{a},\mathbf{b},\mathbf{p}) = \frac{\overrightarrow{\mathbf{ab}} \cdot \overrightarrow{\mathbf{ap}}}{\overrightarrow{\mathbf{ab}} \cdot \overrightarrow{\mathbf{ab}}} \tag{44}$$

$$\mathrm{y}(\mathbf{a},\mathbf{b},\mathbf{p}) = \frac{(\mathcal{R} \cdot \overrightarrow{\mathbf{ab}}) \cdot \overrightarrow{\mathbf{ap}}}{\overrightarrow{\mathbf{ab}} \cdot \overrightarrow{\mathbf{ab}}} \tag{45}$$

$$\mathrm{z}(\mathbf{a},\mathbf{b},\mathbf{p}) = \mathrm{x}^2(\mathbf{a},\mathbf{b},\mathbf{p}) + \mathrm{y}^2(\mathbf{a},\mathbf{b},\mathbf{p}) \tag{46}$$

$$\mathrm{E}(\mathbb{A},\mathbb{B}) = z(\mathbf{a},\mathbf{b},\mathbf{c}) + z(\mathbf{b},\mathbf{a},\mathbf{d}) + z(\mathbf{c},\mathbf{d},\mathbf{a}) + z(\mathbf{d},\mathbf{c},\mathbf{a}) \tag{47}$$

with $\mathbf{a} = \mathbb{A}.\mathbf{p}$, $\mathbf{b} = \mathbb{A}.\mathbf{q}$, $\mathbf{c} = \mathbb{B}.\mathbf{p}$, $\mathbf{d} = \mathbb{B}.\mathbf{q}$, and $\mathcal{R}$ is a rotation by $\pi/2$. Note that $\mathrm{E}(\mathbb{A},\mathbb{B})$ is similarity-invariant. In Fig. 17, we plot it for all 9 PITAs. We observed that, for P456, maximal discrepancy typically occurs near the beginning and/or end of the pattern/animation and that, for P178, maximal discrepancy may be more pronounced and may occur not at an end.

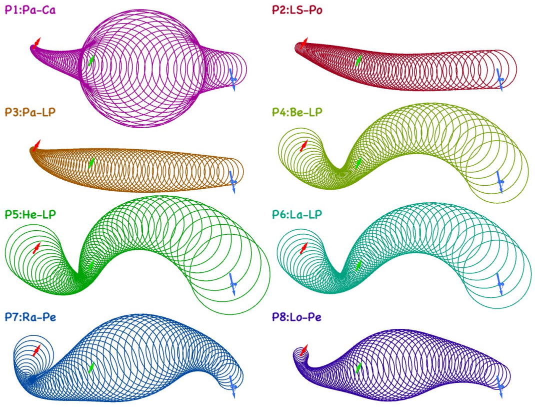


Figure 17: Discrepancy $r = \mathrm{E}(\mathbb{D}(t-0.01), \mathbb{D}(t-0.01))$ shown for each $\mathbb{D}(t)$ as circle of radius $30r$ around $\mathbb{D}(t).\mathbf{p}$.

6.3. Ease of editing

To compare the ease-of-use of the PITAs, we propose the following test. The artist is shown (Fig. 18) a PITA pattern, its keyframes $\mathbb{A}$, $\mathbb{B}$, $\mathbb{C}$, the arrow for $\mathbb{D}(1/4)$, and a target arrow $\mathbb{E}$ and is asked to suggest a new version of $\mathbb{B}$ that would bring $\mathbb{D}(1/4)$ close to $\mathbb{E}$. We measured the number of click&drag ***moves*** needed to align $\mathbb{D}(1/4)$ and $\mathbb{E}$, so that a slight expansion of one covers the other. In our user-interface, if, during the ***click***, the mouse is closer to $\mathbb{B}.\mathbf{p}$ that to $\mathbb{B}.\mathbf{q}$, a move with that ***drags*** by $\vec{\mathbf{v}}$ translates $\mathbb{B}$ by $\vec{\mathbf{v}}$. Otherwise, the move only translates $\mathbb{B}.\mathbf{q}$ by $\vec{\mathbf{v}}$.

Here are results on an informal test. For P123, 2 moves sufficed: the first one aligns $\mathbb{D}(1/4).\mathbf{p}$ and the second one aligns $\mathbb{D}(1/4).\mathbf{q}$. For the other non-independent PITAs, the task is more challenging, because a prior alignment is invalidated by subsequent drags, and because it may not be obvious how to anticipate and compensate for these side effects. For P8, typically 3 or 4 moves sufficed. For P7, we needed between 4 and 13 moves. For P456, we needed between 5 and 8 moves. P5 was often more costly than the other two.

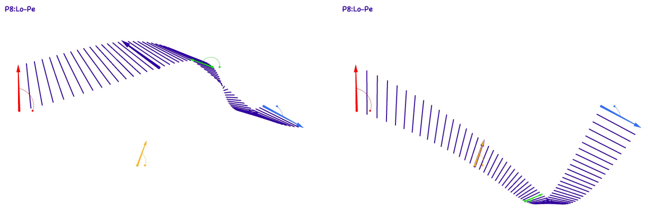


Figure 18: Left: The initial image with frames for P8 and the orange target arrow $\mathbb{E}$. Right: The result after 4 click&drag moves tweaking green $\mathbb{B}$ to try and align $\mathbb{D}(1/4)$ with $\mathbb{E}$.

6.4. Overreaction

P278 are not pop-free (Sec, 4.7), which means that minute tweaks of one of the parameters of the keyframes may cause a jump (discontinuity) in the animation or pattern defined by that PITA. This problem happens near singular configurations. It makes it difficult to edit the animation near these.

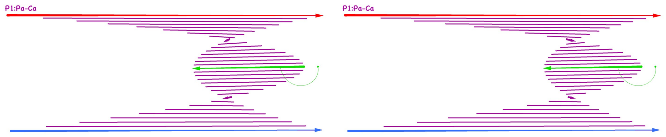


Figure 19: Left: Unstable configuration for P1 for which $\mathbb{D}(1/4).m$ and $\mathbb{D}(3/4).m$ (short arrows) are relatively small. Right: The result of tweaking the tip of $\mathbb{B}$. Note the overreaction of $\mathbb{D}(1/4).w$ and $\mathbb{D}(3/4).w$.

Here, we discuss a softer version of such a pop. A PITA is ***overreacting*** when, in what we call ***unstable configurations***, a minute change of one keyframe parameter may produce a surprisingly large response (change to the path or orientation of some arrows in the pattern/animation). In practice, due to the pixel quantization of the drag, the overreaction of the pattern may appear as a milder pop. P1 may overreact in the orientation of very short frames (Fig. 19). P45 may overreact more dramatically in the path of the animation (Fig. 20).

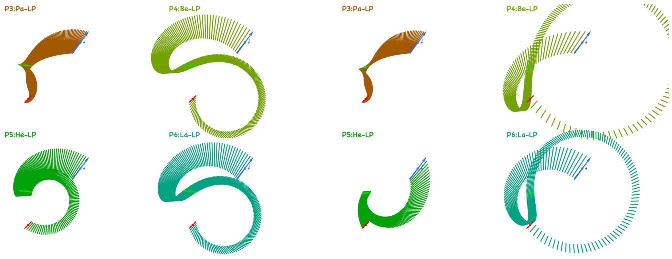


Figure 20: We compare the frame patterns for P3456 before (left) and after (right) a tweak of $\mathbb{B}.\mathbf{q}$. The effect of this tweak is most pronounced for P45. Note that the windings of all keyframes are in $[-\pi, \pi]$.

6.5. Range

Comparing the range of motions or patterns that PITAs may produce is challenging. Here are some notable limitations of some of the chosen PITAs. The curve traced by

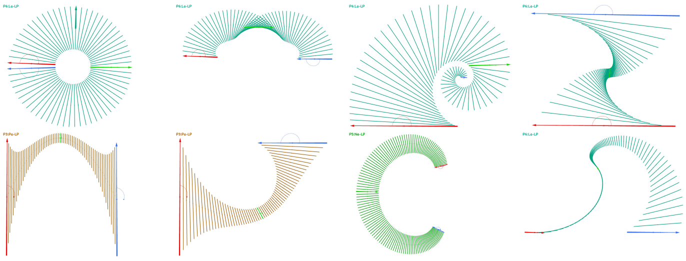


Figure 21: Examples of frame patterns created with different PITAs.

$\mathbb{D}(t).\mathbf{p}$ for P12 is a parabola or a log spiral, and hence has no inflection. P4 does not have this restriction, but, in practice, the curves traced by $\mathbb{D}(t).\mathbf{p}$ and $\mathbb{D}(t).\mathbf{q}$ for P2 rarely exhibit an inflection. P34 are not winding aware, so it is impossible to use them to produce winding motions in which the moving object rotates by more than 2π. The winding motions produced by P1 are rarely satisfying. Nevertheless, our basket of chosen PITAs makes it possible to create a wide variety of motions or patterns (Fig. 21), including perfect circular and log-spiral path and motions with inflection.

Knot-control offered by P1356 extends the range considerably (Fig. 22).

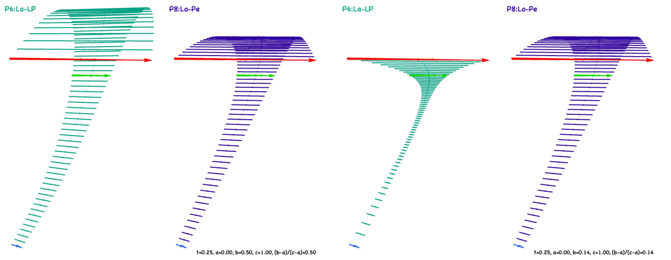


Figure 22: The initial recoil (upward motion) near $\mathbb{A}$ for P68 (left) may be compensated (right), but only for P6, by adjusting the knot-ratio.

6.6. Our recommendations

P1 has many properties that may simplify computation or design, but suffers from severe drawbacks: it is not winding-aware, the path of its $\mathbb{D}(t).\mathbf{p}$ has no inflections, and the behavior of its $\mathbb{D}(t).\vec{\mathbf{w}}$ is inconsistent and difficult to control. Hence, we do not recommend it for most applications.

P23 offer a better control over $\mathbb{D}(t).\vec{\mathbf{w}}$ and P2 offers a nicer path for $\mathbb{D}(t).\mathbf{p}$, but they also have severe drawbacks: the path of their $\mathbb{D}(t).\mathbf{p}$ has no inflections and they are handle-invariant. Furthermore, P2 is not pop-free, not order-independent, and not truly knot-controlled. Hence, we do not recommend them for most applications.

P456 are similar and often produce pleasing animations. However, we noticed that, in ***wild*** configurations with large winding differences and large magnitude ratios, P45 may exhibit excessive bulges and large gains near the start or end of the animation. This makes them difficult to use. Hence, from these three, we recommend P6.

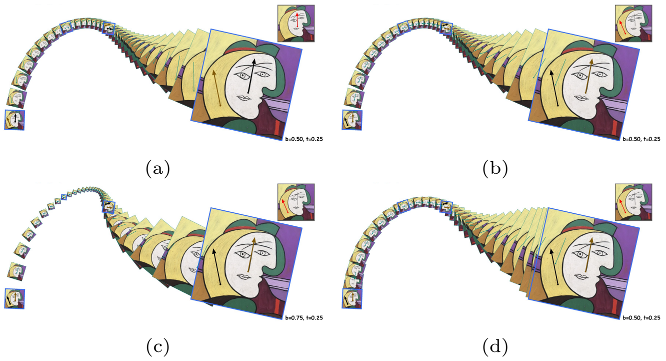
P8 is locally-perseverant, Hence, we expect that it may produce the most pleasing animations. But the patterns that it produces often seem too tight. It is not winding aware and not pop-free. Hence, we recommend it as the best tool for creating pleasing, artifact-free animations in ***tame*** configurations, where the keyframes have modest winding differences and magnitude ratios.

P7 does not create optimal (locally-perseverant) paths for $\mathbb{D}(t).\mathbf{p}$ and does suffer from some of the drawbacks of P8 mentioned above. But its tends to produce less tight motions than P8. We see it as a useful compromise between P6 and P8, at least for tame configurations.

7. Use-case examples

7.1. Designing the motion of an image

Consider an image $\mathbf{X}$. The designer may (1) draw (click&drag) an arbitrary handle-arrow, $\mathbb{E}$, over $\mathbf{X}$ and (2) draw keyframe-arrows $\mathbb{A}$, $\mathbb{B}$, and $\mathbb{C}$, which define keyframe-images, $\mathbf{X}_a$, $\mathbf{X}_b$, and $\mathbf{X}_c$. The moving image, $\mathbf{X}_t$, is defined by $\mathbb{D}(t)$ (Fig. 23). Image/arrow combinations ($\mathbf{X}$, $\mathbb{E}$), ($\mathbf{X}_a$, $\mathbb{A}$), ($\mathbf{X}_b$, $\mathbb{B}$), ($\mathbf{X}_C$, $\mathbb{C}$), and ($\mathbf{X}_t$, $\mathbb{D}(t)$) are similar.



(a) (b) (c) (d)

Figure 23: (a) The animation of picture $\mathbf{X}$ (small insert, top-right) may be designed by placing a ***handle*** (red arrow) over it, and then by specifying its 3 keyframes locations (black), $\mathbb{A}$, $\mathbb{B}$, and $\mathbb{C}$. (b) The same animation may be produced by using a different set of red (handle) and black keyframes. (c) The animation may be tweaked by changing the knot value b. (d) The animation may be softened by a an ease-in/ease-out (time-warp) effect.

The video ***P1268Animation.mov***, submitted with the paper, compares such animations for P1268. Some artifacts are shown in Fig. 24.

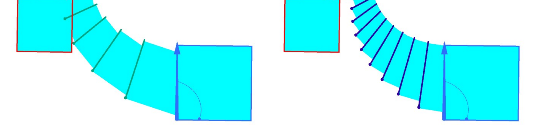


Figure 24: Animation-frames (cyan squares) for P1268. P1 (top-left) shrinks the square excessively. P2 (top-right) exhibits a fast orientation change towards the end (right) of the animation. P6 exhibits a surprising bounce near the $\mathbb{B}$ keyframe (orange square). Using P8 does not alleviate these issues, but does soften them

7.2. *Drawing a variable-width stroke*

By filling the quads defined by the series of consecutive frames in a PITA pattern (Fig. 25), one obtains a tool for designing variable width strokes. We found P456 to be the easiest to use and most effective for this purpose, because, for these, for a useful range of configurations (especially when the keyframes are nearly orthogonal to the centerline), a small change to the length of a keyframe changes the stroke thickness locally in a predictable manner and a small change of the direction of a keyframe changes the centerline of the stroke. These two controls are not as well separated for the other PITAs. We expect that PITAs may be even more useful for representing and animating textured strokes [17, 18].

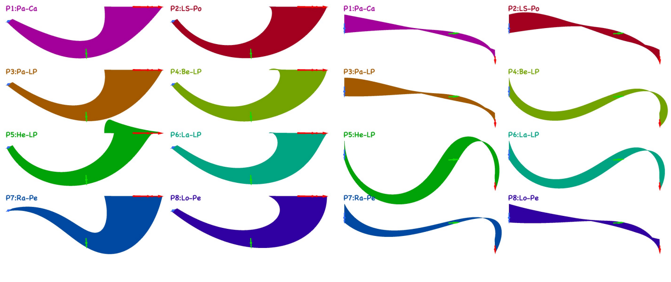


Figure 25: Variable-with strokes defined by the 8 chosen PITAs for two different configurations of keyframes (left and right).

7.3. *Warping a banner*

The start- and end-points, $\mathbb{D}(t).\mathbf{p}$ and $\mathbb{D}(t).\mathbf{q}$, of a pattern of frames (arrow-instances for uniformly spaced time-samples) of a PITA define the vertices of a ***quad-strip*** (Fig. 13), which may be used to bend shapes [19] or texture-mapped and used to display, warp, or animate a banner (Fig. 26). This tool is reasonably straightforward to use and may produce a modest, although useful, set of warps.

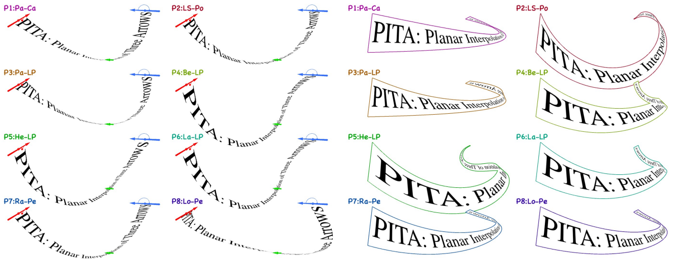


Figure 26: Texture-mapping a text over a quad-strip defined by a PITA may be used to warp a banner. We compare results for all chosen PITAS using two different configurations (left and right) of keyframes.

7.4. *Animating a pattern*

In Fig. 27, we show frames of an animation of a PITA that is controlled by 3 keyframes that move, each being animated by an SMA between two control-arrows (Sec. 2.5). To show it off on a static pictures, we tried to minimize the overlap between consecutive animation frames (instances of the moving pattern of frames). We conjecture that this simple tool might be useful for creating gentle deformations, such as breathing or bulging, or for producing consistent animations of (textured) strokes [20].

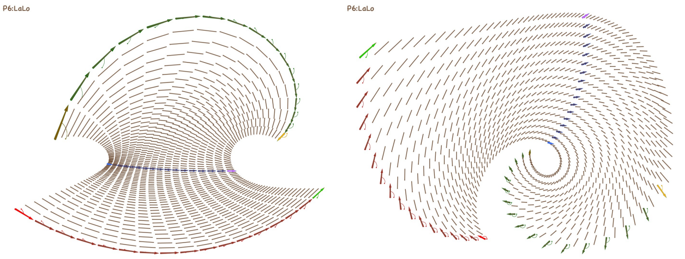


Figure 27: Frames of an animated PITA pattern controlled by 3 arrows (red, blue green), each animated by a Steady Morph of Arrows.

7.5. *Predicting a similarity-motion*

We can use our PITAs to extrapolate the behavior $\mathbb{X}(t)$ of a moving arrow (or object, image, similarity-frame) from three past ***observations***. Specifically, we want to predict $\mathbb{D} = \mathbb{X}(d)$ from $\mathbb{A} = \mathbb{X}(a)$, $\mathbb{B} = \mathbb{X}(b)$, and $\mathbb{C} = \mathbb{X}(c)$. We show all 8 predictors in Fig. 28 for evenly-timed observations. We conjecture that these predictors improve on 2-frame predictors and that they may be useful for predicting motions of components from sensors or videos in a variety of applications [21].

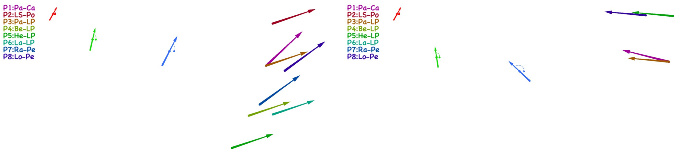


Figure 28: Left: PITA predictors $\mathbb{X}(1.5)$ from 3 observed keyframes, $\mathbb{X}(0)$ (red), $\mathbb{X}(0.5)$ (green), and $\mathbb{X}(1)$ (blue). Right: We snapped $\mathbb{B}$ to $\mathrm{M}(\mathbb{A}, \mathbb{C})$, the Steady Similarity Average of $\mathbb{A}$ and $\mathbb{C}$. In that case, the predictors of P45678 match $\mathrm{M}(\mathbb{B}, \mathbb{C}, 2)$.

7.6. *Interpolating 4 points*

We want a point-motion, $\mathbf{p}(t)$, that interpolates 4 control points: $\mathbf{p}(0) = \mathbf{a}$, $\mathbf{p}(\frac{1}{3}) = \mathbf{b}$, $\mathbf{p}(\frac{2}{3}) = \mathbf{c}$, $\mathbf{p}(1) = \mathbf{d}$.

We propose the following PITA-based solution:

$$\mathbf{X} = \mathrm{Px}(0, \mathbf{ab}, 1/3, \mathbf{bc}, 2/3, \mathbf{cd}, t) \tag{48}$$

$$\mathbf{Y} = \mathrm{Px}(1/3, \mathbf{ab}, 2/3, \mathbf{bc}, 1, \mathbf{cd}, t) \tag{49}$$

$$\mathbf{p}(t) = (1-t)\mathbf{X} + t\mathbf{Y}. \tag{50}$$

For "simple" configurations, all 8 of our PITA-based solutions are acceptable. But for more general configurations, only P178 produce consistently acceptable results, partly because they are not winding-aware. We find the results of P78 more pleasing that those of the centripetal and uniform (P1) Neville interpolations (Fig. 29).

8. Conclusion

We propose, evaluate, and compare 8 PITA solutions to the problem of defining a planar similarity motion that interpolates three given similarity keyframes at specified time-values. We chose these particular solutions because they sometimes yield blatantly different results and because they demonstrate either an adaptation/improvement of a prior solution or a new approach proposed here. We list a set of 11 desirable properties for such interpolating motions and compare which solutions satisfy which of these properties.

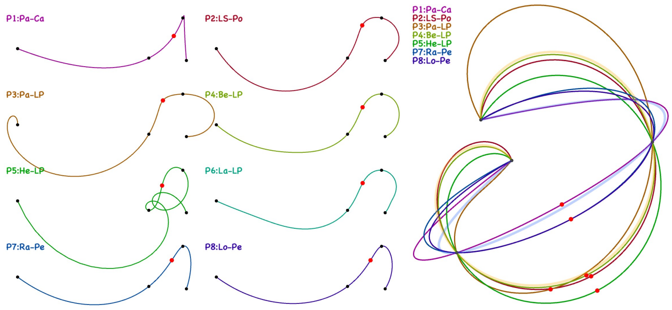


Figure 29: Left: PITA interpolations of 4 points. Right: Comparing them to the centripetal Neville (cyan) and to the log-spiral blend (yellow).

To provide an easy-to-use graphic interface that avoids burdening the user with trigonometric branching problems and that supports the design of a variety of twirling behaviors, we use an extension (which we call twirl) of the concept of a planar similarity. This work shows that the problem of generating a planar motion that interpolates keyframes has many potentially useful solutions. It underscores the central role of steadiness and of perseverance (its extension proposed here). Finally, it demonstrates the challenges posed by trigonometric branching, by the lack of commutativity of some constructions, and by lack of handle-invariance. We show a few examples that illustrate the diversity of potential uses of these PITAs and the disparity of the results they produce. We hope that the observations and techniques presented here will motivate the use of these PITAs and their extensions to affine motions, to more than 3 keyframes, and to three dimensions.

9. Acknowledgements

This research was developed, in part, with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. This work has been partially funded by the Spanish Ministry of Economy and Competitiveness and FEDER Grant TIN2017-88515-C2-1-R. The formulations and implementations for PITA7 (resp. 8) were developed by P. K. Muthumanickam [7], when he was a student at the UPC, in collaboration with Prof. Vinacua (resp. by Zizhen Wang, when he was a student at Georgia Tech, in collaboration with Prof. Rossignac).

References

[1] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, 3rd Edition, Vol. 12 of Texts in Applied Mathematics, Springer-Verlag New York, 2002.

[2] J. Rossignac, À. Vinacua, Steady Affine Motions and Morphs, ACM Transactions on Graphics (TOG) 30 (5) (2011) 116.

[3] J. Rossignac, Corner-Operated Tran-Similar (COTS) maps, patterns, and lattices, ACM Trans. Graph. 39 (1) (Feb. 2020).

[4] K. Kurzeja, J. Rossignac, BeCOTS: Bent Corner-Operated Tran-Similar maps and lattices, Computer-Aided Design (2020) 102912.

[5] A. Gupta, K. Kurzeja, J. Rossignac, G. Allen, P. S. Kumar, S. Musuvathy, Programmed-lattice editor and accelerated processing of parametric program-representations of steady lattices, Computer-Aided Design 113 (2019) 35 – 47.

[6] J. Rossignac, M. Luffel, À. Vinacua, SAMBA: Steadied choreographies, in: Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging, Eurographics, 2012, pp. 1–9.

[7] P. K. Muthumanickam, Uniform accelerated motions, Master's thesis, U.P.C. · Barcelona Tech (September 2012).

[8] A. Haarbach, T. Birdal, S. Ilic, Survey of higher order rigid body motion interpolation methods for keyframe animation and continuous-time trajectory estimation, in: 2018 International Conference on 3D Vision (3DV), 2018, pp. 381–389.

[9] S. Leonardos, C. Allen-Blanchette, J. Gallier, The exponential map for the group of similarity transformations and applications to motion interpolation, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 377–382.

[10] L. Kavan, S. Collins, C. O'Sullivan, J. Žára, Dual quaternions for rigid transformation blending, Tech. Rep. TCD-CS-2006-46, Trinity College, Dublin (2006).

[11] T. W. Sederberg, P. Gao, G. Wang, H. Mu, 2-D shape blending: An intrinsic solution to the vertex path problem, in: SIGGRAPH, ACM, 1993, p. 15–18.

[12] M. Alexa, Linear combination of transformations, in: J. Hughes (Ed.), SIGGRAPH, ACM, 2002, pp. 380–387.

[13] F. C. Park, B. Ravani, Smooth invariant interpolation of rotations, ACM Trans. Graph. 16 (3) (1997) 277–295.

[14] J.-D. Fekete, É. Bizouarn, É. Cournarie, T. Galas, F. Taillefer, Tictactoon: A paperless system for professional 2d animation, in: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, 1995, pp. 79–90.

[15] B. Whited, G. Noris, M. Simmons, R. W. Sumner, M. Gross, J. Rossignac, Betweenit: An interactive tool for tight inbetweening, Computer Graphics Forum 29 (2) (2010) 605–614.

[16] S. Wirtz, D. Paulus, Evaluation of established line segment distance functions, Pattern Recognit. Image Anal. 26 (2) (2016) 354–359.

[17] M. P. Salisbury, M. T. Wong, J. F. Hughes, D. H. Salesin, Orientable textures for image-based pen-and-ink illustration, in: SIGGRAPH, ACM, 1997, p. 401–406.

[18] M. Bessmeltsev, J. Solomon, Vectorization of line drawings via polyvector fields, ACM Trans. Graph. 38 (1) (Jan. 2019).

[19] Y. Lipman, V. G. Kim, T. A. Funkhouser, Simple formulas for quasiconformal plane deformations, ACM Trans. Graph. 31 (5) (Sep. 2012).

[20] G. Noris, D. Sýkora, S. Coros, B. Whited, M. Simmons, A. Hornung, M. Gross, R. W. Sumner, Temporal noise control for sketchy animation, in: ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering, NPAR '11, ACM, 2011, p. 93–98.

[21] E. Barsoum, J. Kender, Z. Liu, Hp-gan: Probabilistic 3d human motion prediction via gan, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2018, pp. 1418–1427.

[22] F. Park, B. Ravani, Bézier curves on riemannian manifolds and lie groups with kinematics applications, Journal of Mechanical Design 117 (1995) 36–40.

[23] V. I. V. I. Arnold́, Ordinary Differential Equations, MIT Press, Cambridge, MA, 1973, translated and edited by Richard A. Silverman.

Appendix A. Log and Exp of linear and homogeneous forms

The homogeneous matrix, $H(\mathbb{A})$, associated with arrow $\mathbb{A} = \mathrm{A}(\mathbf{p}, m, w)$ is:

$$H(\mathbb{A}) = \begin{bmatrix} m\cos(w) & -m\sin(w) & \mathbf{p}_x \\ m\sin(w) & m\cos(w) & \mathbf{p}_y \\ 0 & 0 & 1 \end{bmatrix} \tag{A.1}$$

Its linear part, $L(\mathbb{A})$, is:

$$L(\mathbb{A}) = \begin{bmatrix} m\cos(w) & -m\sin(w) \\ m\sin(w) & m\cos(w) \end{bmatrix} \tag{A.2}$$

The log of $L(\mathbb{A})$ may be computed [9] using

$$\log(L(\mathbb{A})) = \begin{bmatrix} \log(m) & -w \\ w & \log(m) \end{bmatrix} \tag{A.3}$$

Let $L = x\log(L(\mathbb{A})) + y\log(L(\mathbb{B})) + z\log(L(\mathbb{C}))$. The exponential of L is defined as the infinite sum

$$e^L = \exp\left(\begin{bmatrix} a & -\alpha \\ \alpha & a \end{bmatrix}\right) = \sum_{i=0}^{\infty} \frac{L^i}{i!} \tag{A.4}$$

and may be computed [9] using

$$e^L = \begin{bmatrix} e^a\cos(\alpha) & -e^a\sin(\alpha) \\ e^a\sin(\alpha) & e^a\cos(\alpha) \end{bmatrix} = e^a R_\alpha \tag{A.5}$$

To compute the log and exponential of homogenous matrices corresponding to similarities, notice that if

$$M = \left[\begin{array}{cc|c} \multicolumn{2}{c|}{L} & \vec{\mathbf{v}} \\ \hline 0 & 0 & 0 \end{array}\right]$$

where L is a 2 matrix and $\vec{\mathbf{v}}$ is a 2D vector, then

$$M^n = \left[\begin{array}{cc|c} \multicolumn{2}{c|}{L^n} & L^{n-1}\vec{\mathbf{v}} \\ \hline 0 & 0 & 0 \end{array}\right]$$

So if L is as in (A.4) and we set $\vec{\mathbf{v}} = L\vec{\mathbf{t}}$, we obtain from (A.4) and (A.5) that

$$e^M = \left[\begin{array}{cc|c} \multicolumn{2}{c|}{e^a R_\alpha} & (e^a R_\alpha - I)\vec{\mathbf{t}} \\ \hline 0 & 0 & 1 \end{array}\right] \tag{A.6}$$

so using (A.1), (A.3) and (A.6) we see that the

$$\log(H(\mathbb{A})) = \begin{bmatrix} \log(m) & -\alpha & \mathbf{q}_x \\ \alpha & \log(m) & \mathbf{q}_y \\ 0 & 0 & 0 \end{bmatrix}, \tag{A.7}$$

where

$$\mathbf{q} = \begin{bmatrix} \log(m) & -\alpha \\ \alpha & \log(m) \end{bmatrix} (mR_w - I)^{-1}\,\mathbf{p} \tag{A.8}$$

which is more straightforward than the computation in [9]. Notice that the matrix $(mR_w - I)^{-1}$ may be singular only if $w = k\pi$ for some integer k, and if moreover $m = 1$, so $H(\mathbb{A}) = I$ and there is no motion at all.

Given three arrows, we can now form $H = x\log(H(\mathbb{A})) + y\log(H(\mathbb{B})) + z\log(H(\mathbb{C}))$. This new matrix H will have the same shape as the logs of a similarity (i.e. the first two columns and rows form a matrix of the form $\lambda I + S$ for some skew-symmetric matrix S, and the last row is filled with zeros), so we can use (A.6) to compute its exponential.

Observe also that the linear part of H is identical to e^L defined above.

Appendix B. P3:Pa-LP versus P10:Pa-EL

In this appendix, we compare P3:Pa-LP to P10:Pa-EL, a non-chosen PITA, which uses the parabolic interpolant of P3 for $\mathbb{D}(t).\mathbf{p}$, but, instead of using the approach proposed here for the -LP solutions, computes $\mathbb{D}(t).\vec{\mathbf{w}}$ using the exponent of the Weighted Arithmetic Mean of the logarithms of the linear parts of the similarity transformations associated with the keyframes (see Appendix A). P10 is not winding-aware, as shown in Fig. B.30. It may be made winding-aware by using a logic to correctly select the proper branch of the exponential. With that improvement, the result produced by P10 would be identical to the one produced by P3. Hence, P3 improves on P10 in two ways: (1) It uses a much simpler expression that avoids the complication and possible numeric rounding errors of computing matrix logarithms and exponentials, and (2) it does not require a post-processing logic to select the proper trigonometric branch.

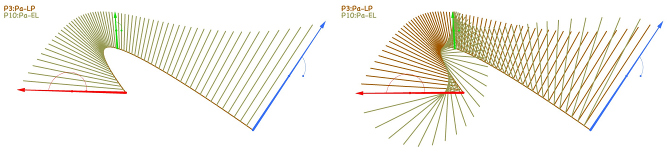


Figure B.30: P3:Pa-LP and P10:Pa-EL, for which $\mathbb{D}(t).\vec{\mathbf{w}}$ is computed using a weighted arithmetic average of the logs of the linear parts of the three similarities). Left: Identical results. Right: A tweak of $\mathbb{A}.w$ produces a pop of P10:Pa-EL.

Appendix C. P5: Similarity-invariance

Prior art [22] proposes to define the moving arrow $\mathbb{H}(t)$ in terms of the corresponding homogeneous matrix computed as $H(t) = \exp(x\log(H(\mathbb{A})) + y\log(H(\mathbb{A})) + z\log(H(\mathbb{C})))$.

This solution is not affected by rotations or dilations about the global origin, $\mathbf{g}$. By this, we mean that rotating (resp. dilating) the keyframes about the origin and then computing $\mathbb{H}(t)$ yields the same result as computing $\mathbb{H}(t)$ first and then performing that rotation (resp. dilation).

Unfortunately, this solution is affected by translation. Hence, it is not similarity invariant. For example, rotating the keyframe configuration about the centroid $\mathbf{o}$ of the origins of the keyframes distorts the pattern.

Furthermore, the above solution is not handle-invariant, which is a considerable drawback for animation design.

We use the following heuristic to address this problem: (1) translate the keyframes by $\overrightarrow{\mathbf{og}}$, (2) compute $H(t)$, as explained above, (3) extract the corresponding arrow, $\mathbb{D}(t)$, and (4) translate it by $\overrightarrow{\mathbf{go}}$. We use this solution for PITA P5:He-LP. It is similarity invariant.

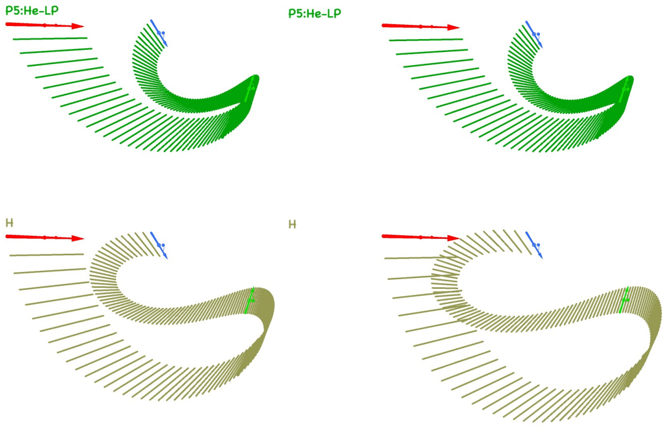


Figure C.31: Initial patterns (left) of P5 (top) and of $\mathbb{H}(t)$ (bottom) for the same keyframes. Result (right) after translating the keyframes to the right.

Still, two issues remain: (1) Another heuristic (such as defining $\mathbf{o}$ as the centroid of the end points of the keyframe arrows, or as $\mathbb{A}.\mathbf{p}$ yields a different $\mathbb{D}(t)$) and (2) P5 is not handle-invariant (Fig. C.31).

Appendix D. P6 versus other -LP candidates

In this appendix, we compare P6:La-LP to 20 other options that we explored and explain why we decided not to include these in our basket of chosen PITAs. We split them into variants of the Lagrange and of the Neville formulations.

Appendix D.1. Lagrange Variants

We considered a variant of Eq. 34:

$$\mathrm{T}_2(a, \mathbb{A}, b, \mathbb{B}, c, \mathbb{C}, t) = \mathrm{T}(\mathrm{M}(\mathbb{A}, \mathbb{B}, y), \mathbb{A}, \mathbb{C}, z) \tag{D.1}$$

where y and z are the ratios defined in Eq. 10.

For each variant, T_1 and T_2, we consider 6 permutations of the timed-keyframes.

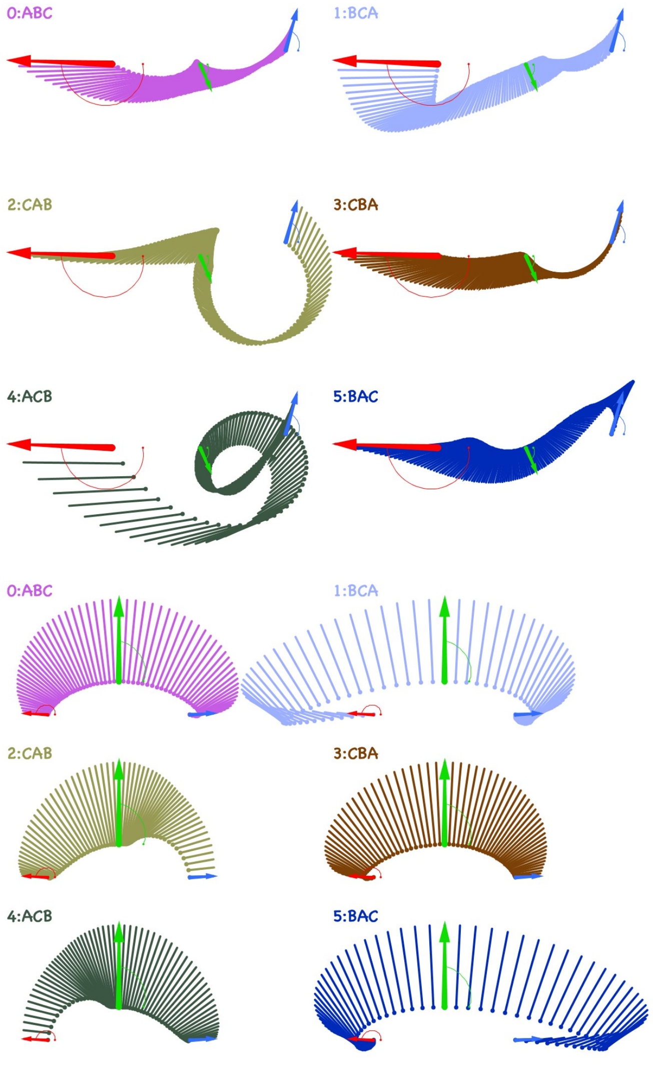


Figure D.32: Top: The 6 T_1 permutations. Bottom: Same but for a symmetric configuration of keyframes.

Let $\mathrm{T}_{1\mathbb{ABC}}$ stand for $\mathrm{T}_1(a, \mathbb{A}, b, \mathbb{B}, c, \mathbb{C}, t)$. It transforms $\mathbb{A}$ by the SMA from $\mathbb{A}$ to $\mathbb{B}$ and then by the STA from $\mathbb{B}$ to $\mathbb{C}$. Similarly, let $\mathrm{T}_{2\mathbb{ABC}}$ stand for $\mathrm{T}_2(a, \mathbb{A}, b, \mathbb{B}, c, \mathbb{C}, t)$. Each T_1 is identical to a different T_2 variant: $\mathrm{T}_{1\mathbb{ABC}} = \mathrm{T}_{2\mathbb{BAC}}$, $\mathrm{T}_{1\mathbb{BCA}} = \mathrm{T}_{2\mathbb{CBA}}$, $\mathrm{T}_{1\mathbb{CAB}} = \mathrm{T}_{2\mathbb{ACB}}$, $\mathrm{T}_{1\mathbb{CBA}} = \mathrm{T}_{2\mathbb{BCA}}$, $\mathrm{T}_{1\mathbb{ACB}} = \mathrm{T}_{2\mathbb{CAB}}$, and $\mathrm{T}_{1\mathbb{BAC}} = \mathrm{T}_{2\mathbb{ABC}}$. For example, $\mathrm{T}_{2\mathbb{BAC}} = \mathcal{T}^{z}_{\mathbb{B},\mathbb{C}} \cdot \mathcal{T}^{x}_{\mathbb{B},\mathbb{A}} \cdot \mathbb{B} = \mathcal{T}^{z}_{\mathbb{B},\mathbb{C}} \cdot \mathcal{T}^{1-y-z}_{\mathbb{B},\mathbb{A}} \cdot \mathbb{B} = \mathcal{T}^{z}_{\mathbb{B},\mathbb{C}} \cdot \mathcal{T}^{y+z}_{\mathbb{A},\mathbb{B}} \cdot \mathcal{T}^{1}_{\mathbb{B},\mathbb{A}} \cdot \mathbb{B}$ $= \mathcal{T}^{z}_{\mathbb{B},\mathbb{C}} \cdot \mathcal{T}^{y+z}_{\mathbb{A},\mathbb{B}} \cdot \mathbb{A} = \mathrm{T}_{1\mathbb{ABC}}$. Hence, we only consider the 6 T_1 variants (Fig. D.32-left). Furthermore, $\mathrm{T}_{1\mathbb{ABC}}$ is the time-reverse of $\mathrm{T}_{1\mathbb{CBA}}$, $\mathrm{T}_{1\mathbb{BCA}}$ of $\mathrm{T}_{1\mathbb{BAC}}$, and $\mathrm{T}_{1\mathbb{CAB}}$ of $\mathrm{T}_{1\mathbb{ACB}}$ (Fig. D.32-right). So, we consider only the averages (which are symmetric) of these 3 pairs.

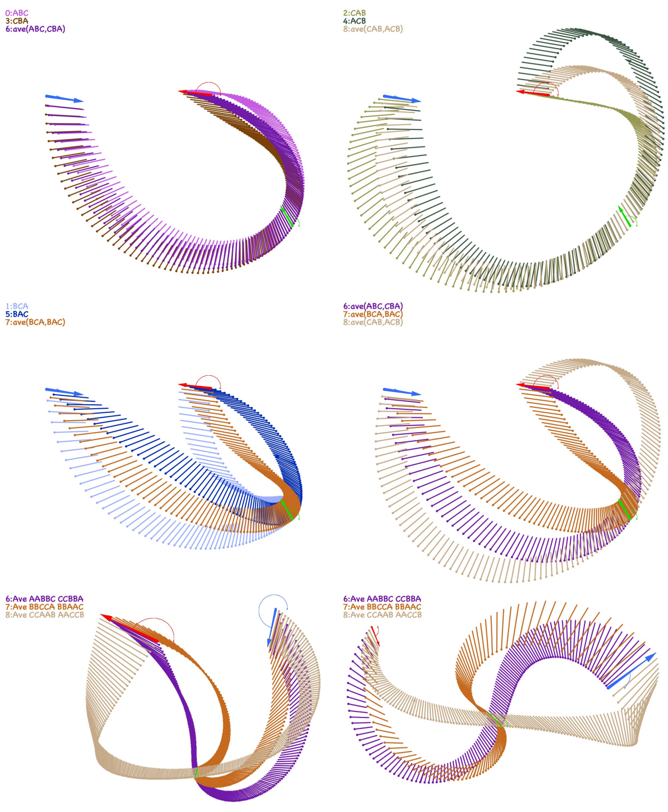


Figure D.33: Top-left: $\mathrm{T}_{1\mathbb{ABC}}$, $\mathrm{T}_{1\mathbb{CBA}}$, and their average. Top-right: $\mathrm{T}_{1\mathbb{BCA}}$, $\mathrm{T}_{1\mathbb{BAC}}$, and their average. Middle-left:$\mathrm{T}_{1\mathbb{CAB}}$, $\mathrm{T}_{1\mathbb{ACB}}$, and their average. Remaining: comparing the 3 averages (P6 is in magenta).

We chose, as P6, the average of $\mathrm{T}_{1\mathbb{ABC}}$ and $\mathrm{T}_{1\mathbb{CBA}}$, because it offers a compromise between the other two (Fig. D.33).

Appendix D.2. P9:Ne-LP and other Neville Variants

Before inventing P6, we considered ***P9:Ne-LP***, which is an SMA of two SMAs (***Neville*** form):

$$\mathrm{M}(a, \mathbb{A}, b, \mathbb{B}, c, \mathbb{C}, t) = \mathrm{M}(\mathrm{M}(\mathbb{A}, \mathbb{B}, q), \mathrm{M}(\mathbb{B}, \mathbb{C}, r), s) \tag{D.2}$$

using the SMA (Eq. 20) and ratios in Eq. 14.

We compare it here to options obtained by changing the order of the arrows in this construction, and discuss two variants obtained by replacing the SMA building block, which P5 uses, by a non-steady morph.

Construction: The three SMA steps in the construction of P9 are illustrated in Fig. D.34.

Other options: We define other options for P9 through permutations of the three timed keyframes: $(a, \mathbb{A})$, $(b, \mathbb{B})$, and $(c, \mathbb{C})$. For conciseness, we use the following notation:

$$\mathrm{M}_{\mathbb{ABC}} = \mathrm{M}(a, \mathbb{A}, b, \mathbb{B}, c, \mathbb{C}, t). \tag{D.3}$$

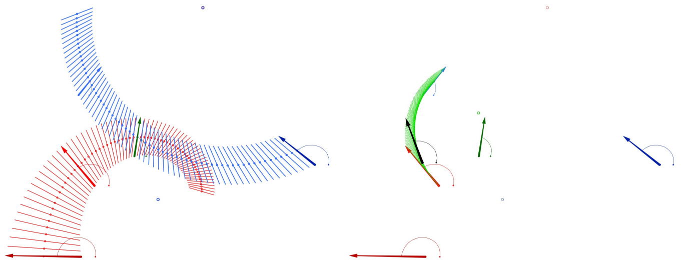


Figure D.34: Given $\mathbb{A}$ (red), $\mathbb{B}$ (green), $\mathbb{C}$ (blue), we show (left) the patterns for timed SMAs, $\mathrm{M}(a, \mathbb{A}, b, \mathbb{B}, t)$ (dark-red) and $\mathrm{M}(b, \mathbb{B}, c, \mathbb{C}, t)$ (dark-blue). Both are extended to show the frames for time-range $[a, c]$. We highlight the corresponding arrows, $\mathbb{L}$ (red) and $\mathbb{R}$ (blue), for $t = a + (a - c)/4$. We use these as keyframes (right), to define a timed SMA $\mathrm{M}(a, \mathbb{L}, c, \mathbb{R}, t)$ (green frames), and highlight (black arrow) its frame for $t = a + (a - c)/4$, which is what P9 returns.

These 6 permutations have the following properties:

- $\mathrm{M}_{\mathbb{ABC}} = \mathrm{M}_{\mathbb{CBA}}$, $\mathrm{M}_{\mathbb{ACB}} = \mathrm{M}_{\mathbb{BCA}}$, $\mathrm{M}_{\mathbb{BAC}} = \mathrm{M}_{\mathbb{CAB}}$
- $\mathrm{M}_{\mathbb{ABC}}$ is symmetric
- $\mathrm{M}_{\mathbb{ACB}}$, and $\mathrm{M}_{\mathbb{BAC}}$ are the ***time-reverse*** of each other: swapping $(a, \mathbb{A})$ and $(c, \mathbb{C})$ swaps $\mathrm{M}_{\mathbb{ACB}}$ and $\mathrm{M}_{\mathbb{BAC}}$

In Fig. D.35 , we show that $\mathrm{M}_{\mathbb{ABC}}$, $\mathrm{M}_{\mathbb{ACB}}$, and $\mathrm{M}_{\mathbb{BAC}}$ can be very different and that $\mathrm{M}_{\mathbb{ABC}}$ is a good compromise between $\mathrm{M}_{\mathbb{ACB}}$ and $\mathrm{M}_{\mathbb{BAC}}$, and is close to their average. Hence, we use it as P9.

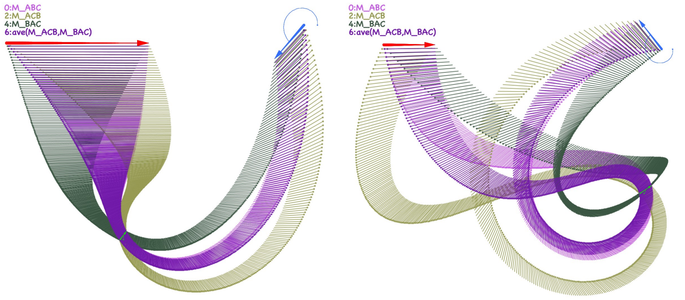


Figure D.35: $\mathrm{M}_{\mathbb{ABC}}$, $\mathrm{M}_{\mathbb{ACB}}$, $\mathrm{M}_{\mathbb{BAC}}$, and the average of $\mathrm{M}_{\mathbb{ACB}}$ and $\mathrm{M}_{\mathbb{BAC}}$ superimposed for two configurations. Notice the similarity between that average and $\mathrm{M}_{\mathbb{ABC}}$.

In Fig. D.36, we compare P9 to variants (Sec. 2.7) obtained by replacing WGM by WAM or WHM for both the computation of the path and of the magnitude of the moving arrow.

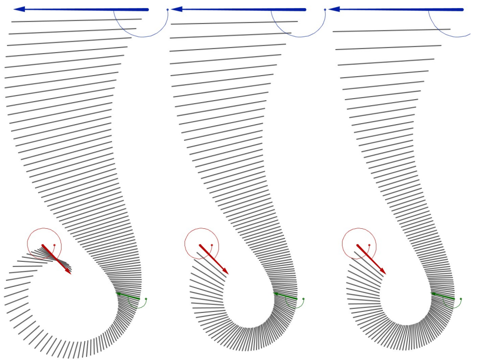


Figure D.36: We show 3 variants using WAM (left), WGM (center), and WHM (right) for the path of the center of the arrow and for its length. P9 (WGM) is a compromise of the other two.

P9 and P6 produce very similar results for configurations where the orientations of the keyframes are similar (Fig. D.37-left). We chose to include P6, and not P9, in the chosen basked of PITAs because, in some configurations (see for example Fig. D.37-right), P9 may significantly deviate from P6 near one or both ends of the pattern in ways that we find unnatural.

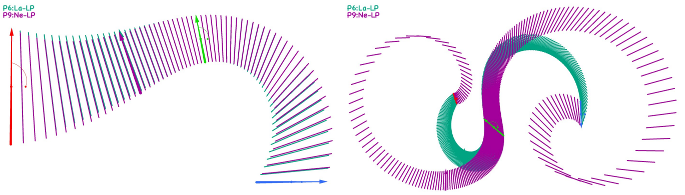


Figure D.37: Comparing P6 (aquamarine) and P9 (berries) for a tame (left) and more challenging (right) configurations.

Appendix E. P7:Ra-Pe (Details)

In P7, the acceleration of a point, $\mathbf{p}$, is a constant similarity transform of its displacement, $\overrightarrow{\mathbf{fp}}$, from a fixed point $\mathbf{f}$. Hence, the trajectory of the two end-points of $\mathbb{D}(t)$ and, in fact of any point transported by P4, satisfies

$$\mathbf{p}''(t) = \mathcal{S} \cdot \overrightarrow{\mathbf{fp}}(t) \tag{E.1}$$

where $\mathcal{S}$ is a 2×2 matrix, representing the combination of a dilation and a rotation. We rewrite this equation to form a first order differential equation

$$\mathbf{z}'(t) = \mathcal{M} \cdot \mathbf{z}(t), \tag{E.2}$$

where

$$\mathbf{z}(t) = \begin{bmatrix} \overrightarrow{\mathbf{fp}}(t) \\ \mathbf{p}'(t) \end{bmatrix}, \qquad \mathcal{S} = \begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix} \tag{E.3}$$

$$\mathcal{M} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ c_1 & c_2 & 0 & 0 \\ c_3 & c_4 & 0 & 0 \end{bmatrix} \tag{E.4}$$

which has solution

$$\mathbf{z}(t) = e^{t\mathcal{M}} \cdot \mathbf{z}(0). \tag{E.5}$$

Because $\mathcal{S}$ is the linear part of a similarity transform, the four eigenvalues of $\mathcal{M}$ have the following structure

$$\begin{aligned} \lambda_1 &= \alpha + \beta i, \quad \lambda_2 = -\alpha - \beta i \\ \lambda_3 &= \alpha - \beta i, \quad \lambda_4 = -\alpha + \beta i \end{aligned}$$

for some real numbers α and β, and the matrix composed by the corresponding eigenvectors is

$$\mathcal{V} = \begin{bmatrix} \vec{\mathbf{v}}_1 \\ \vec{\mathbf{v}}_2 \\ \vec{\mathbf{v}}_3 \\ \vec{\mathbf{v}}_4 \end{bmatrix}^T = \begin{bmatrix} (-\alpha i - \beta)/k & (\alpha - \beta i)/k & -ki & k \\ (\alpha i + \beta)/k & (-\alpha + \beta i)/k & -ki & k \\ (\alpha i - \beta)/k & (\alpha + \beta i)/k & ki & k \\ (-\alpha i + \beta)/k & (-\alpha - \beta i)/k & ki & k \end{bmatrix}^T$$

where $k = \sqrt{\alpha^2 + \beta^2}$. The inverse of this matrix is

$$\mathcal{V}^{-1} = \frac{1}{4} \begin{bmatrix} (\alpha i - \beta)/k & (\alpha + \beta i)/k & i/k & k \\ (-\alpha i + \beta)/k & (-\alpha - \beta i)/k & i/k & k \\ (-\alpha i - \beta)/k & (\alpha - \beta i)/k & -i/k & k \\ (\alpha i + \beta)/k & (-\alpha + \beta i)/k & -i/k & k \end{bmatrix}$$

We can compute $e^{t\mathcal{M}}$ using

$$e^{t\mathcal{M}} = \mathcal{V} \begin{bmatrix} e^{\lambda_1 t} & 0 & 0 & 0 \\ 0 & e^{\lambda_2 t} & 0 & 0 \\ 0 & 0 & e^{\lambda_3 t} & 0 \\ 0 & 0 & 0 & e^{\lambda_4 t} \end{bmatrix} \mathcal{V}^{-1}, \tag{E.6}$$

and combining equations E.1, E.3, E.5 and E.6, the final closed-form solution for the trajectory of the point is

$$\mathbf{p}(t) = \mathbf{f} + \mathcal{S}_1(t) \cdot \overrightarrow{\mathbf{fp}}(0) + \mathcal{S}_2(t) \cdot \mathbf{p}'(0), \tag{E.7}$$

where $\mathcal{S}_1(t)$ is $\begin{bmatrix} s_{cc}(t) & -s_{ss}(t) \\ s_{ss}(t) & s_{cc}(t) \end{bmatrix}$, $\mathcal{S}_2(t)$ is

$$\frac{1}{\alpha^2+\beta^2} \begin{bmatrix} \alpha s_{sc}(t) + \beta s_{cs}(t) & -\alpha s_{cs}(t) + \beta s_{sc}(t) \\ \alpha s_{cs}(t) - \beta s_{sc}(t) & \alpha s_{sc}(t) + \beta s_{cs}(t) \end{bmatrix},$$

$\mathbf{p}'(0)$ is the initial velocity of $\mathbf{p}$, and

$$\begin{array}{ll} s_{cc}(t) = \cosh(\alpha t)\cos(\beta t), & s_{ss}(t) = \sinh(\alpha t)\sin(\beta t) \\ s_{cs}(t) = \cosh(\alpha t)\sin(\beta t), & s_{sc}(t) = \sinh(\alpha t)\cos(\beta t) \end{array}.$$

Notice that both $\mathcal{S}_1(t)$ and $\mathcal{S}_2(t)$ are antisymmetric, so they represent some time-varying similarity transform. The unknowns to be solved for are α, β, the fixed point $\mathbf{f}$, a point $\mathbf{p}(0)$ and its initial velocity $\vec{\mathbf{v}}_0$. These define 8 degrees of freedom, which matches the number of constraints.

We denote the three zero-turn arrows to interpolate by $\mathbf{p}_a\mathbf{q}_a$, $\mathbf{p}_b\mathbf{q}_b$, and $\mathbf{p}_c\mathbf{q}_c$. For symmetry, we set their knots to -1, 0 and 1. Using the properties that $\mathcal{S}_1(1) = \mathcal{S}_1(-1)$ and $\mathcal{S}_2(1) = -\mathcal{S}_2(-1)$, the values of $s_{ss}(1)$ and $s_{cc}(1)$ can be obtained by solving the following system of equations

$$\overrightarrow{\mathbf{p_a q_a}} + \overrightarrow{\mathbf{p_c q_c}} = 2 \begin{bmatrix} s_{cc}(1) & -s_{ss}(1) \\ s_{ss}(1) & s_{cc}(1) \end{bmatrix} \overrightarrow{\mathbf{p_b q_b}}.$$

Then the values of $\sinh(\alpha)$ can be computed using

$$\sinh(\alpha) = \mathrm{sgn}(s_{ss}(1)) \sqrt{\frac{w + \sqrt{w^2 + 4s_{ss}^2}}{2}},$$

where $w = s_{ss}(1) + s_{cc}(1) - 1$.

Then the values of α and β are computed using:

$$\alpha = \sinh^{-1}\left(\mathrm{sgn}(s_{ss}(1)) \sqrt{\frac{w + \sqrt{w^2 + 4s_{ss}^2}}{2}} \right), \tag{E.8}$$

$$\beta = \cos^{-1}\left(\frac{s_{cc}(1)}{\sqrt{\sinh(\beta)^2 + 1}} \right). \tag{E.9}$$

We use these values to compute the initial velocities for $\mathbf{p}$ and $\mathbf{q}$ (i.e., the velocities of the start-point, $\mathbb{D}(t).\mathbf{p}$, and the end-point, $\mathbb{D}(t).\mathbf{q}$, of the moving arrow $\mathbb{D}(t)$ at time $t = b = 0$, when $\mathbb{D}(t)$ interpolates keyframe $\mathbb{B}$):

$$\mathbf{p}'(0) = \frac{1}{2}\mathcal{S}_2(1)^{-1} \cdot \overrightarrow{\mathbf{p_a p_c}} \tag{E.10}$$

$$\mathbf{q}'(0) = \frac{1}{2}\mathcal{S}_2(1)^{-1} \cdot \overrightarrow{\mathbf{q_a q_c}} \tag{E.11}$$

and the fixed point, $\mathbf{f}$, of P7:

$$\mathbf{f} = \mathbf{p}_b + (\mathcal{S}_1(1) - \mathcal{I})^{-1} \cdot \left(\mathbf{p}_b - \frac{\mathbf{p}_a + \mathbf{b}_c}{2} \right) \tag{E.12}$$

where $\mathcal{I}$ is the identity matrix. Observe that $\mathbf{f} = \mathbf{p}_b$ when $\mathbf{p}_b = \frac{\mathbf{p}_a + \mathbf{b}_c}{2}$.

Appendix F. P8:Lo-Pe (Details)

This PITA is based on the ***Uniformly Accelerated Motion*** (UAM). We give here a succinct derivation, and further details may be found in [7].

Consider a point trajectory $\mathbf{p}(t) = \mathcal{U}(t) \cdot \mathbf{p}_0$, where $\mathcal{U}(t)$ is an affinity. At time t, the point moves with an acceleration $\frac{d^2}{dt^2}\mathbf{p}(t)$ which can be obtained by applying the second derivative of $\mathcal{U}(t)$ to point $\mathbf{p}_0$. Then the condition that the acceleration be constant along each trajectory in the local reference frame amounts to all six non-zero entries of the homogeneous matrix of the affinity $\mathcal{U}''(t)$ to be constant in that reference frame (see Eq. (39)). Expanding the matrix, one obtains six second order ODEs in the six unknown entries of $\mathcal{U}$. These can be combined to obtain separate systems of equations for pairs of entries in this matrix, all of the same form. One such system is

$$\begin{array}{l} \frac{d^2}{dt^2}a(t) = C_1 a(t) + C_2 b(t) \\ \frac{d^2}{dt^2}b(t) = C_3 a(t) + C_4 b(t) \end{array} \tag{F.1}$$

where $a(t)$ and $b(t)$ are the entries of the solution matrix in the first two columns of the first row, and the C_i are arbitrary constants (that represent the choice of constant acceleration).

This second order ODE is transformed in the usual way [23] into a first order system

$$\mathcal{Z}'(t) = \mathcal{M} \cdot \mathcal{Z}(t),$$

where

$$\mathcal{M} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ C_1 & C_2 & 0 & 0 \\ C_3 & C_4 & 0 & 0 \end{bmatrix} \text{ and } \mathcal{Z}(t) = \begin{bmatrix} a(t) \\ b(t) \\ a'(t) \\ b'(t) \end{bmatrix}$$

which can then be solved straightforwardly as

$$\mathcal{Z}(t) = e^{(t\mathcal{M})} \cdot \mathcal{Z}(0). \tag{F.2}$$

A similar system is satisfied by $d(t)$ and $e(t)$, the first two entries in the second row of the matrix $\mathcal{U}$, completing the first two columns of the affinity matrix. The remaining entries corresponding to the translation part can be found by plugging into the original equation the solutions for the first four coefficients, obtaining a second order differential equation on these last two entries of the matrix. In [7] the reader may find a detailed derivation of the closed-form expressions for the exponential of the matrix $\mathcal{M}$, and of the translation part of the affinity in terms of the boundary conditions $\mathcal{U}(0) = \mathcal{A}$, $\mathcal{U}(\frac{1}{2}) = \mathcal{B}$ and $\mathcal{U}(1) = \mathcal{C}$.

From the standard ODE theory, one sees that P8 is **unique**. It is also **affine invariant**, and in the form used here (interpolating affinities at $t = \{0, \frac{1}{2}, 1\}$), it is symmetric. If all three affinities $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are similarities, then $\mathcal{U}(t)$ is also a similarity for all t.