ScrewBender: Polyscrew Subdivision for Smoothing Interpolating Motions

Alexander Powell[†‡] and Jarek Rossignac[§]

College of Computing, Georgia Institute of Technology

Abstract

An ordered series of control poses may be interpolated by a polyscrew rigid body motion composed of a series of screws, each interpolating a pair of consecutive control poses. The trajectory of each point during screw motion is (C^∞). Although a polyscrew is continuous, velocities are typically discontinuous at control poses when the motion switches between screws. We obtain a smooth motion by subdividing the polyscrew. Three subdivision schemes are proposed: the interpolating 4-point subdivision, the smooth cubic B-spline subdivision, and the Jarek subdivision, an average of these two. Their implementation is trivial and their computation sufficiently fast for realtime subdivision and animation, which is particularly important for interactive motion editing and hardware support for animation.

Categories and Subject Descriptors (according to ACM CCS): G.1.1 [Interpolation]: Spline and piecewise polynomial interpolation; I.3.5 [Computational Geometry and Object Modeling]: Geometric Transformations; I.3.7 [Three-Dimensional Graphics and Realism]: Animation

1. Introduction

A rigid object whose position and orientation evolve with time is undergoing a **motion**. At each moment, t, its position and orientation are given by a **pose** L(t). The pose may be specified in terms of a rigid body transformation that maps a local coordinate system L, in which the object was defined, into a current coordinate system L(t) defined by an origin O(t) and by three orthonormal basis vectors I(t), J(t), and K(t). At a time t, a vertex of the object defined by its three coordinates (x, y, z) in L will be at O(t)+xI(t)+yJ(t)+zK(t). The time-evolving origin O(t) and basis vectors I(t), J(t), and K(t) are typically specified in terms of a discrete sequence of user-defined or captured **control poses**.

For simplicity, we ignore dynamic constraints and assume that the sequence of control poses are sampled at uniformly spaced parameter values. If this was not the case, a polynomial mapping $p(t)$ between parameter t and time in an animation could be used to resynchronize the motion yielding $\mathrm{O}(p(t))+x\mathrm{I}(p(t))+y\mathrm{J}(p(t))+z\mathrm{K}(p(t))$. For instance, one may specify constraints in value $p(i) = p_i$ and derivative $p'(i) = v_i$ of p at the times initially associated with control poses. These constraints can be interpolated by a C^1 piecewise cubic map.

A motion interpolating two control poses, $\mathrm{L}_0 = [\mathrm{O}_0, \mathrm{I}_0, \mathrm{J}_0, \mathrm{K}_0]$ and $\mathrm{L}_1 = [\mathrm{O}_1, \mathrm{I}_1, \mathrm{J}_1, \mathrm{K}_1]$, is a pose-valued function L(t) satisfying the constraints $\mathrm{L}(0) = \mathrm{L}_0$ and $\mathrm{L}(1) = \mathrm{L}_1$. If L($t$) is defined as a combination of a minimum angle rotation with a linear translation from O_0 to O_1, the result will depend on the choice of the position and orientation of the local coordinate system L, with respect to the object. Such a dependency may create surprising and unwanted effects [RK01]. To overcome this problem, we use screw motions to interpolate between poses. A screw is fully defined by the initial and final control poses. It combines a minimum-angle linear rotation around a fixed axis of direction S with a minimum-distance linear translation along S.

Interpolating each pair of consecutive poses of a control cycle by a screw yields a **polyscrew** motion. Although con-

† e-mail: alexander@gatech.edu

‡ now at DreamWorks Animation

§ e-mail: jarek@cc.gatech.edu

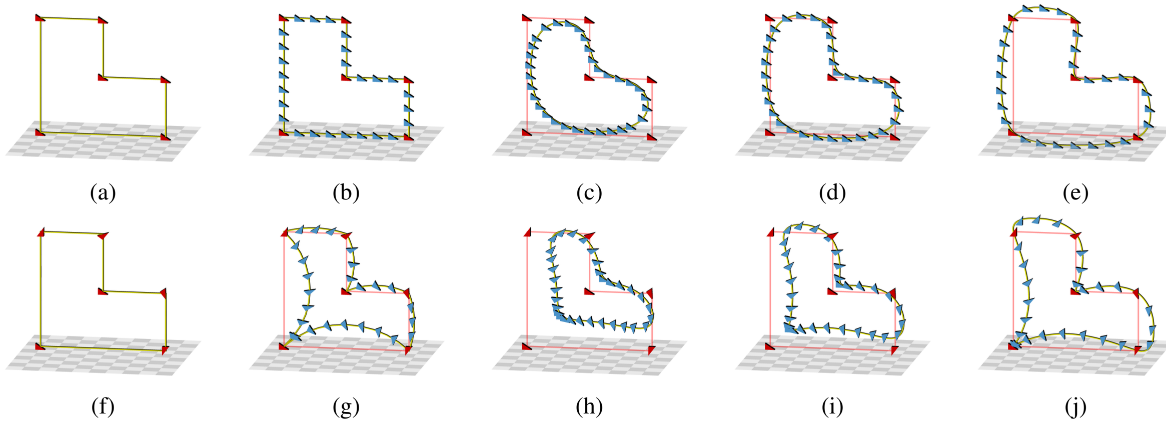


Figure 1: A cycle of parallel control poses defines an animation with sharp changes of velocity at control poses (a). The trajectory of a polygonal translation interpolating them (b) may be smoothed by a cubic B-spline (c), Jarek (d), or 4-point polygon subdivision (e). When four of the control poses are rotated (f), a polyscrew that interpolates between consecutive poses by independent screw motions (g) may be smoothed by the proposed ScrewBender approach, which adapts to polyscrews the Split&Tweak formulations of the cubic B-spline (h), Jarek (i) and 4-point (j) subdivisions. Note that in spite of the apparent cusp in the trajectories at the bottom-left control pose, the resulting animations are smooth as the object decelerates at the cusp. The three subdivided motions are at least C^1. The B-spline is C^2.

tinuous, polyscrews may exhibit sharp discontinuities of velocity at control poses, where the motion switches between one screw and the next one. The **ScrewBender** subdivision proposed here may be used to smooth polyscrews. It is based on the realization that the Split&Tweak formulation of polygon-smoothing subdivision techniques [Ros04] may be trivially extended to polyscrews. Hence, starting from an initial **control polyscrew** defined by the original control poses, a Split&Tweak polyscrew subdivision step produces a refined polyscrew with twice as many control poses. As this subdivision process is repeated, the refined polyscrew converges quickly to a close approximation of a smooth motion that concatenates short screw motions. The polyscrew subdivision and the polyscrew animations are simple and may be computed in realtime. Hence ScrewBender is well suited for the interactive design of motions (Figure 10).

We also assume that the control poses form a **control cycle** and that the motion is cyclic. Acyclic motions defined by a non-cyclic **sequence** of control poses may be easily supported by extending them with dummy screws, treating them as cyclic motions, and removing from the subdivided motion 3 spans for the B-spline or 5 spans for the other schemes.

2. Smoothing Translational Motions

A closed loop control **polygon** of n vertices may be subdivided into a polygon of $2n$ vertices by a subdivision process that inserts new vertices and possibly displaces the old ones. Several subdivision schemes have been suggested (Figure 2). The uniform cubic B-spline curve defined by the control polygon may be approached by a subdivision process [Sab02] that inserts a new vertex in the middle of each edge and moves each old vertex b in a sequence (a,b,c) of consecutive vertices to $(a+6b+c)/8$. Cubic B-spline curves are attractive because they are C^2 smooth. Unfortunately, they do not interpolate the original control vertices. The 4-point subdivision scheme [NDG87] inserts a new vertex at $(-a+9b+9c-d)/16$ between each pair of consecutive vertices b and c in a sequence (a,b,c,d) and leaves the old vertices in place, hence ensuring that the subdivided curve interpolates the original vertices. The Jarek subdivision [Ros04] uses, at each step, for each old and new vertex, the average of its positions as computed by the other two schemes. It offers a compromise between B-spline and 4-point and nearly preserves area of the control polygon in 2D.

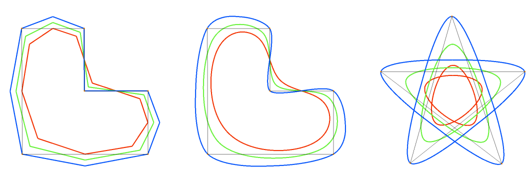


Figure 2: The original L-shaped control polygon (gray) is subdivided once (left) and then 5 times (center) by the B-spline (red), by the 4-point (blue), and by the Jarek (green) schemes. Corresponding subdivisions of a more complex polygon is shown (right).

When all the control poses of an animation are obtained by pure translation from L (no rotation), a smooth animation may be defined by considering the origins O_i of the consecutive poses to be the vertices of a control polygon and by refining this polygon through a B-spline, Jarek, or 4-point subdivision (Figure 1, top row). This solution is simple, fast, and independent of the choice of L. Unfortunately, it is limited to pure translations.

3. Polyscrew Composition and Animation

The representation of a polyscrew is an array of control poses, each represented by the origin O_i and by the associated basis $[I_i, J_i, K_i]$.

In order to support rotations between control poses, we interpolate each pair of consecutive poses by a screw, which combines a translation by dS, where S is a unit vector, with a rotation by angle r around an axis parallel to S through a point P. Consider poses, $L_0 = [O_0, I_0, J_0, K_0]$ and $L_1 = [O_1, I_1, J_1, K_1]$. Following [RK01], to obtain the parameters of the interpolating screw, let $I = I_1 - I_0$, $J = J_1 - J_0$, $K = K_1 - K_0$, and $O = O_1 - O_0$. At least one of the three cross-products $I \times J$, $J \times K$, and $K \times I$ is not a null vector. Without loss of generality, assume that $I \times J$ is the longest of the three (otherwise, rotate the symbols). Compute $S = I \times J$ and normalize it with $S = S/\|S\|$. Compute r as the angle between $S \times I_0$ and $S \times I_1$. Compute d as the dot product $O \cdot S$. Finally, compute $P = (O_1 + O_2 + (S \times O)/\tan(r/2))/2$.

To animate the motion of an object along a single screw, we vary parameter t, translate L_0 by tdS and rotate it by angle tr around the screw axis through P with direction S. As t varies from 0 to 1, $L(t)$ follows a screw interpolating between L_0 to L_1. (These transformations are supported in hardware on commodity graphics adapters.) Note that we can move along the screw past L_0 or L_1 by simply letting t fall outside [0,1].

The resulting polyscrew motion is continuous, but usually not smooth at the control poses (Figure 1g).

4. Smoothing Polyscrew Motions

Our contribution is simply to combine these two ideas (Split&Tweak and polyscrews) and define subdivision rules for polyscrews that correspond to the cubic B-spline, Jarek, and 4-point subdivisions of polygons. ScrewBender provides this capability. It only requires the ability to compute and evaluate screws interpolating pairs of control poses, using the simple procedure reviewed above.

ScrewBender is based on the Split&Tweak reformulation [Ros04] of the three subdivision rules discussed earlier. At each step, all three subdivisions first perform a split operation that inserts a new vertex in the middle of each edge. They then each perform a tweak operation. The B-spline tweak moves the old vertices halfway towards the average of their new neighbors. The 4-point tweak moves the new vertices by one-quarter away from the average of their new second-degree neighbors. The Jarek tweak moves the old vertices by half of the B-spline tweak and the new vertices by half of the 4-point tweak displacements (Figure 3).

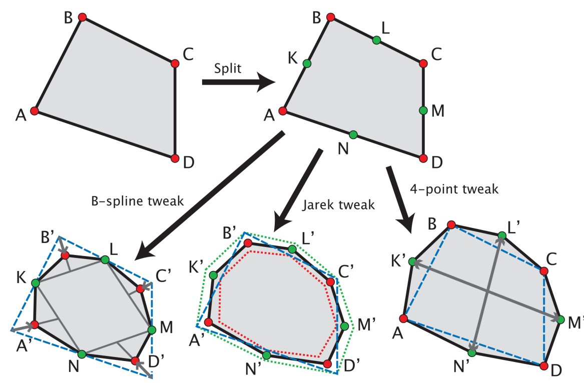


Figure 3: The polygon (top-left) is split by inserting a new vertex in the middle of each edge. Then, vertices are tweaked to complete the B-spline, Jarek, or 4-point subdivision step.

Adopting the Split&Tweak formulation to polyscrews is simple. Let the procedure $s(L_0, t, L_1)$ return a pose $L_{0,1}(t)$ obtained by moving L_0 by a fraction t of the screw motion defined by L_0 and L_1. For example, $L_{0,1}(0) = L_0$ and $L_{0,1}(1) = L_1$. To compute $s(L_0, t, L_1)$, we first compute the screw parameters S, r, d, and P, as discussed in Section 3, and then translate L_0 by tdS and rotate it by angle tr around the screw axis through P with direction S.

For the B-spline scheme, we insert a new control pose $L_{a,b}(\frac{1}{2})$ between each pair of consecutive poses L_a and L_b and obtain a new cycle of control poses alternating between old and new poses. This corresponds to the split step and does not change the motion. Then, we replace each old pose L_b in a subsequence (L_a, L_b, L_c) by $s(s(L_b, \frac{1}{4}, L_a), \frac{1}{2}, s(L_b, \frac{1}{4}, L_c)))$.

For the 4-point scheme, we insert a new control pose computed by $s(s(L_a, \frac{9}{8}, L_b), \frac{1}{2}, s(L_d, \frac{9}{8}, L_c)))$ between control poses L_b and L_c in the subsequence (L_a, L_b, L_c, L_d).

For the Jarek scheme, compute (but do not insert) new poses using $s(s(L_a, \frac{17}{16}, L_b), \frac{1}{2}, s(L_d, \frac{17}{16}, L_c)))$ between control poses L_b and L_c in the subsequence (L_a, L_b, L_c, L_d). Then, we replace each old pose L_b in a subsequence (L_a, L_b, L_c) by $s(s(L_a, \frac{7}{8}, L_b), \frac{1}{2}, s(L_c, \frac{7}{8}, L_b)))$. Finally, we insert the new poses.

These steps are illustrated in Figure 4 and the results are shown in Figures 1(h-j).

5. Results and Applications

In this section, we demonstrate how both simple and complex smooth motions can be easily specified using ScrewBender.

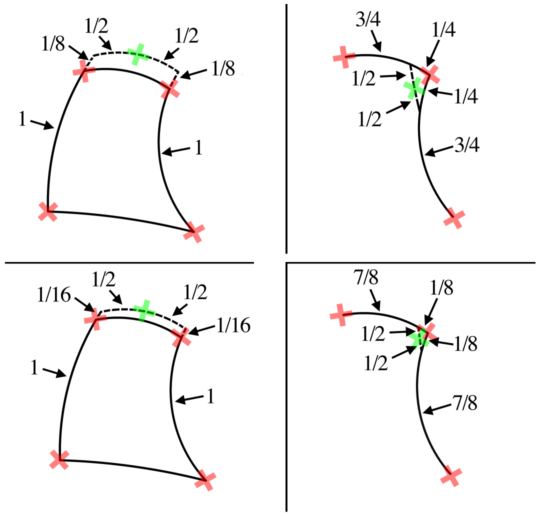


Figure 4: An example is used to demonstrate the Split&Tweak process for polyscrew subdivision. Top left: A new pose is inserted and tweaked in 4-point subdivision. Top right: After new poses are inserted in B-spline subdivision, an old pose is tweaked. The Jarek subdivision process involves tweaking the new vertices (bottom left) and tweaking the old (bottom right).

The smoothly subdivided L-shaped polyscrew of Figure 1(j) may be edited by dragging or rotating a graphically selected control pose (Figure 10). The new control polyscrew is subdivided in realtime and rendered using the stroboscopic instancing, which acts as a 3D rubber-band to provide convenient visual feedback. Our non-optimized implementation can compute more than 110,000 $s(\mathrm{L}_a, t, \mathrm{L}_b)$ operations per second on a 1.5 GHz Apple PowerBook G4.

Simple smooth motions can be easily defined using a few poses as demonstrated by Figure 5.

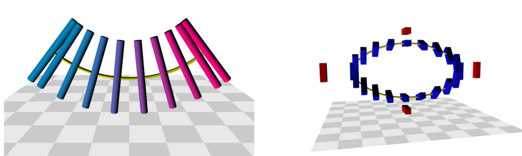
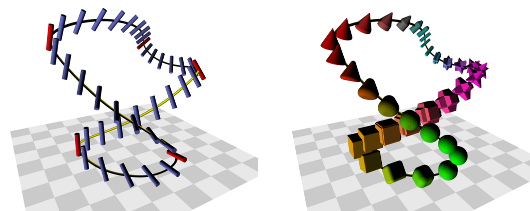


Figure 5: A polyscrew obtained through 4-point subdivision from only two control poses produces a smooth pendulum motion (left). A polyscrew obtained through B-spline subdivision from four control poses (in red) produces a spinning motion of a block (right).

It takes only 5 poses to define the complex, yet smooth, motion in Figure 6a. Note that interpolating motions may be trivially combined with automatic PIP 3D morphing techniques [KR92]. When the interpolated shapes are sufficiently simple, the combined PIPs and ScrewBender subdivision and animation may still be performed in realtime (Figure 6b).



(a) (b)

Figure 6: A stroboscopic rendering of the motion of a cylinder (a). In (b), a different shape is associated with each control pose of the animation. As the object moves along the 4-point subdivided polyscrew, its shape morphs between a cone, a star, a block, a cylinder, and a sphere.

Sometimes, the degree of smoothness of a motion may be derived by analyzing the fairness of the curve that represents it in the Lie group SE(3) [PR97]. Due to the lack of tools for performing such an analysis on our polyscrew refinement scheme, we can only offer conclusions on continuity properties suggested through a numeric test, as was done by Hofer, Pottmann, and Ravani [HPR02] for showing that their pose refinement scheme converges to a C^2 motion. As they did, we use discrete plots of positions, velocity, and acceleration of points to explore the smoothness of the limit motion produced by the three polyscrew Split&Tweak schemes described here. We include one example in Figure 7. From studying these plots for a variety of motions, we conjecture that the 4-point polyscrew subdivision yields C^1 trajectories for points on the model and that the B-spline polyscrew subdivision yields C^2 trajectories. The Jarek subdivision yields at least C^1 trajectories.

6. Contributions in the Context of Prior Art

The problem of computing smooth motions that interpolate a series of control poses has received a considerable amount of attention from the graphics and animation communities. An overview may be found in [Ale02].

Shoemake [Sho85, SD92] animated transformations by using quaternions interpolated with spherical linear interpolation (SLERP). While screw motions have velocity and angular velocity in the direction of the screw axis, SLERP only has an angular velocity component, necessitating the addition of a stretch matrix interpolated in matrix space in Shoemake's interpolated transformations. Barr et al. [BCGH92] demonstrated smooth interpolation of orientations with angular constraints using quaternions. Kim et al. [KKS95] formulated a general framework for unit quaternion splines. These unit quaternion splines are computationally intensive. Möller and Hughes [MH99] computed efficiently a matrix

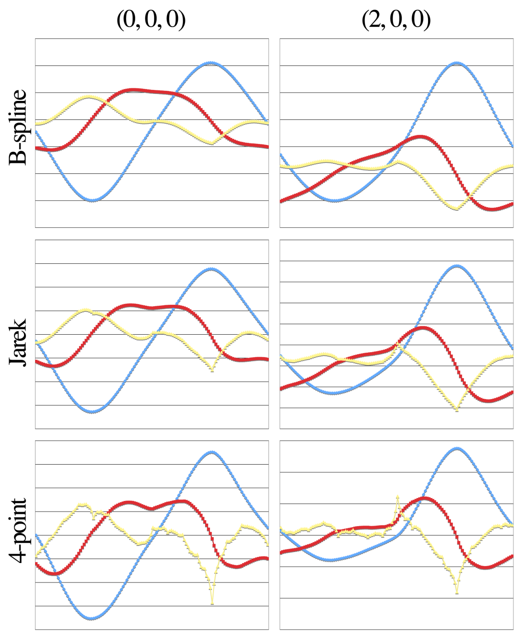


Figure 7: Plots of the x-coordinate of the points (0,0,0) (left column) and (2,0,0) (right column) on the moving object (blue), of its first derivative (red), and of its second derivative (gold) as a function of time for the B-spline, Jarek, and 4-point polyscrew subdivisions.

that would rotate one vector to another, but without consideration of smoothness from one rotation to the next. Grassia [Gra98] introduced to computer graphics the concept of using an exponential map for parameterizing rotations. Finally, Alexa's [Ale02] interpolation in Lie groups by means of the exponential map was extended to handle general transformations.

In particular, we expect that the motion blending used by Alexa [Ale02] to compute intermediate poses between two consecutive control poses could also be used to implement our $s(\mathrm{L_a}, t, \mathrm{L_b})$ procedure, and hence the Split&Tweak subdivisions. In fact, Alexa has suggested that his scheme could be used to implement a Bezier and 4-point subdivision. The computational cost of such a solution as discussed in [Ale02] and the fact that the result will be dependent on the choice of a coordinate system lead us to believe that ScrewBender offers an interesting alternative for the realtime design of simple smooth motions.

It is possible to decouple the motion into a translation and a rotation around the center of the object. The translation can be smoothed through 4-point polygon subdivision. The rotation may be smoothed through a restricted version of the polyscrew subdivision or through subdividing or blending spherical spline curves on the S^3 unit sphere [BF01] as proposed by several authors [Sho85, Sho87, Duf86, WJ93, RBG88, KN95]. Such an approach has two disadvantages with respect to the non-decoupled polyscrew subdivision advocated here. First, the dependency on the choice of the "center" of the object may produce undesirable artifacts [JW96], especially when the moving object is changing shape (Figure 6b). Second, the motion may sometimes appear less natural. For example, when the poses are sampled along a linear translation, along a circular motion, or along a screw motion, all three of our Split&Tweak polyscrew subdivisions will preserve the initial trajectory perfectly (Figure 8a). However, a decoupled smoothing, where the center of the object follows a trajectory obtained by a 4-point polyloop subdivision and where the orientation is smoothed as described here, does not preserve the initial trajectory (Figure 8b).

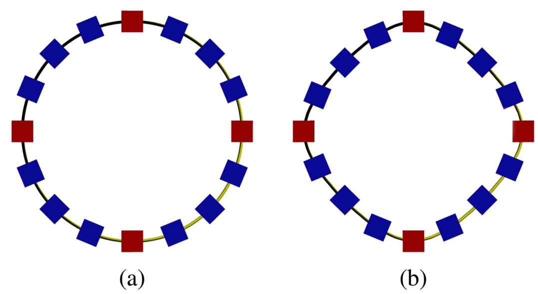


Figure 8: When subdivided with any of the polyscrew subdivisions presented here, a circular motion defined by four control poses (shown in red) retains its initial trajectory (a). A decoupled smoothing results in a less natural subdivided trajectory (b).

Hofer, Pottmann, and Ravani [HPR02] offer two motion refinement approaches, both of which refine the motion by iteratively inserting intermediate poses. The resulting dense set of discrete poses may be sufficient for most applications or, if a continuous motion is desired, it may be interpolated by piecewise screw motions without concern for smoothness.

Their first approach applies the global, linear variational subdivision scheme proposed by Kobbelt [Kob96] to four arbitrary non-coplanar feature points on the object. Because the scheme is linear, the position of all points at an intermediate pose may be derived from these four through linear combination. The result is an affinely distorted object. They use a least-square registration [Hor87] to best align the moving shape with its distorted copy, and hence produce a new intermediate pose for the rigid body motion. They invoke the smoothness of discrete plots of the x-coordinate of a point on the object and of its first derivative as evidence that the motion is C^2.

Their second approach computes the translations of the intermediate poses using the above global variational subdivision to refine the positions of the local origin. They compute the orientation of each pose using a local angle-minimizing interpolation, which corresponds to our $s(\mathrm{A}, \frac{1}{2}, \mathrm{B})$ primitive described in detail above. Hence, the inserted poses are not the mid-screw poses produced by the Split stage of our approach. Instead, they correspond to a decoupled scheme where the translation part is computed independently from the rotation. Then, they correct all of the new poses using linearized motions. To each feature point, they attach its velocity vector computed assuming a piecewise screw motion interpolating the neighboring poses. Finally, they minimize the variations of these velocities by solving a system of linear equations, whose solution defines screw-motion tweaks to the intermediate poses. Again, discrete plots suggest that the resulting motion is C^2.

Finally, Wallner and Pottmann [WP06] offer an angle-minimizing helical interpolation and show that the C^1 and C^2 smoothness of certain geodesic subdivision methods extend to helical subdivision in the Lie group SE(3).

7. Conclusion

We have proposed an extremely simple approach for the construction of smooth motions defined by a cycle of control poses. It is based on the ScrewBender subdivision scheme, which adapts the cubic B-spline, Jarek, and 4-points polygon subdivision schemes to polyscrews. ScrewBender can subdivide and animate complex polyscrews in realtime. We have combined it with automatic 3D morphing to provide a simple and effective tool for the interactive design of animations where 3D objects move and morph.

The simplicity and speed of the approach proposed here makes it a useful alternative to the large number of previously proposed schemes, even though some of them may yield higher order continuity.

By coupling the rotational and translational components into a piecewise screw motion we achieve natural interpolations that perfectly preserve linear, circular, and screw trajectories. The generality of the Split&Tweak formulation permits the creation of a large variety of motions with very few control poses.

The user may insert control poses anywhere in the sequence and interactively rotate and translate any control pose with real-time feedback provided by the stroboscopic rendition. Because a stroboscopic rendition of a complex subdivided polyscrew can be computed and rendered at more than 20 frames per second while manipulating the initial control poses, ScrewBender offers the ideal environment for the direct manipulation of motions.

References

[Ale02] ALEXA M.: Linear combination of transformations. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (2002), ACM Press, pp. 380–387.

[BCGH92] BARR A. H., CURRIN B., GABRIEL S., HUGHES J. F.: Smooth interpolation of orientations with angular velocity constraints using quaternions. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques* (1992), ACM Press, pp. 313–320.

[BF01] BUSS S. R., FILLMORE J. P.: Spherical averages and applications to spherical splines and interpolation. *ACM Trans. Graph. 20*, 2 (2001), 95–126.

[Duf86] DUFF T.: Splines in animation and modeling. In *SIGGRAPH '86 Course Notes on State of the Art Image Synthesis* (1986), ACM Press.

[Gra98] GRASSIA F. S.: Practical parameterization of rotations using the exponential map. *J. Graph. Tools 3*, 3 (1998), 29–48.

[Hor87] HORN B. K. P.: Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Amer. 4*, 4 (April 1987), 629–642.

[HPR02] HOFER M., POTTMANN H., RAVANI B.: Subdivision algorithms for motion design based on homologous points. In *Advances in Robot Kinematics: Theory and Applications* (2002), Lenarcic J., Thomas F., (Eds.), Kluwer Academic Publishers, pp. 235–244.

[JW96] JUTTLER B., WAGNER M.: Computer aided design with spatial rational b-spline motions. *ASME Journal of Mechanical Design*, 118 (1996), 193–201.

[KKS95] KIM M.-J., KIM M.-S., SHIN S. Y.: A general construction scheme for unit quaternion curves with simple high order derivatives. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), ACM Press, pp. 369–376.

[KN95] KIM M. S., NAM K. W.: Interpolating solid orientations with circular blending quaternion curves. *Computer Aided Design 27* (1995), 385–398.

[Kob96] KOBBELT L.: Interpolatory subdivision on open quadrilateral nets with arbitrary topology. *Comput. Graph. Forum 15*, 3 (1996), 409–420.

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

[MH99] MÖLLER T., HUGHES J. F.: Efficiently building a matrix to rotate one vector to another. *J. Graph. Tools 4*, 4 (1999), 1–4.

[NDG87] N. DYN D. L., GREGORY J. A.: A four-point interpolatory subdivision scheme for curve design. *Computer Aided Design 4* (1987), 257–268.

[PR97] PARK F. C., RAVANI B.: Smooth invariant interpolation of rotations. *ACM Trans. Graph. 16*, 3 (1997), 277–295.

[RBG88] ROBERTS K. S., BISHOP G., GANAPATHY S. K.: Smooth interpolation of rotational matrices. In *Proceedings CVPR '88: Computer Vision and Pattern Recognition* (1988), IEEE Computer Science Press, pp. 724–729.

[RK01] ROSSIGNAC J., KIM J. J.: Computing and visualizing pose-interpolating 3d motions. *Computer Aided Design 33*, 4 (2001), 279–291.

[Ros04] ROSSIGNAC J.: Education-driven research in cad. *Computer Aided Design 36*, 14 (2004), 1461–1469.

[Sab02] SABIN M.: Subdivision surfaces. In *The Handbook of Computer-Aided Geometric Design*, G. Farin J. H., Kim M. S., (Eds.). 2002, ch. 12, pp. 309–327.

[SD92] SHOEMAKE K., DUFF T.: Matrix animation and polar decomposition. In *Proceedings of the conference on Graphics interface '92* (1992), Morgan Kaufmann Publishers Inc., pp. 258–264.

[Sho85] SHOEMAKE K.: Animating rotation with quaternion curves. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques* (1985), ACM Press, pp. 245–254.

[Sho87] SHOEMAKE K.: Quaternion calculus and fast animation. In *SIGGRAPH '87 Course Notes on State of the Art Image Synthesis* (1987), ACM Press, pp. 101–121.

[WJ93] WANG W., JOE B.: Orientation interpolation in quaternion space using spherical biarcs. In *Graphics Interface '93* (May 1993), pp. 24–32.

[WP06] WALLNER J., POTTMANN H.: Intrinsic subdivision with smooth limits for graphics and animation. *ACM Trans. Graph. 25*, 2 (2006), 356–374.

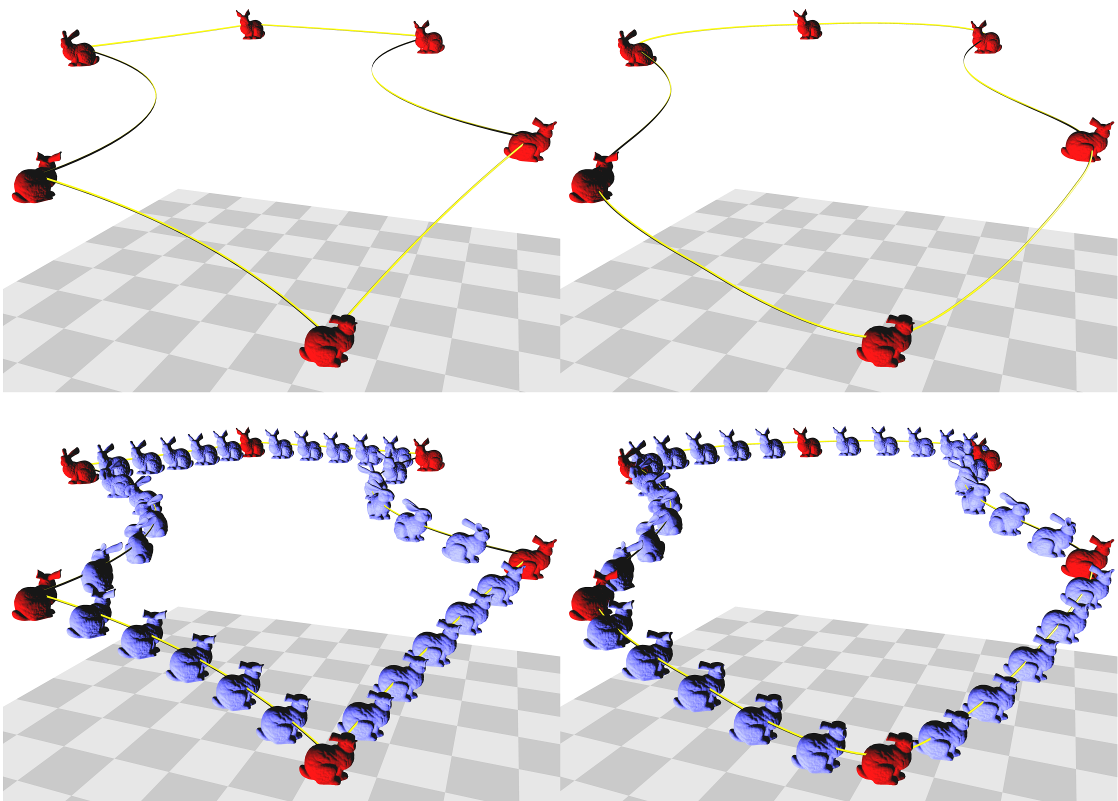


Figure 9: A cyclic animation of a non-planar motion of bunny is defined with 6 control poses (shown in red). Top left: The path for an animation composed of single screws that each interpolate two consecutive control poses. Notice the sharp direction discontinuities at the control poses. Bottom left: Intermediate poses for the above trajectory, showing the orientation of the bunny. Top Right: The path of an animation that interpolated the same control poses produced by our ScrewBender 4-point polyscrew subdivision. Notice that the trajectory is now smooth. Bottom right: Intermediate poses for the smooth animation.

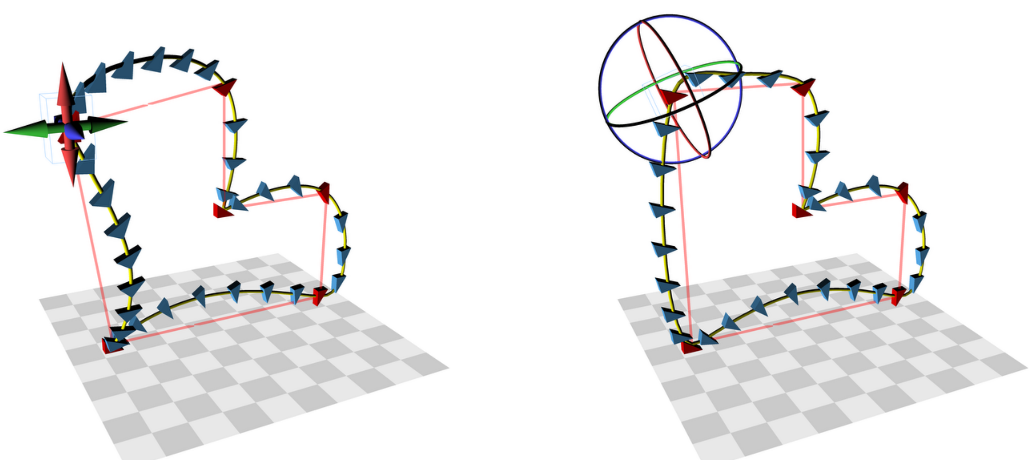


Figure 10: The stroboscopic rendering of the subdivided polyscrew in Figure 1 is updated in realtime as the user translates (left) or rotates (right) the top-left control pose of the model. Note that the user has local control over the smooth motion.