

ScrewBender: Smoothing Piecewise Helical 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 (piecewise helical) rigid body motion composed of a series of screws, each interpolating a pair of consecutive control poses. The trajectory of each point during a screw motion segment 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 new 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.

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

Keywords: motion, screw, smoothing, subdivision

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 $O(p(t))+xI(p(t))+yJ(p(t))+zK(p(t))$. 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, $L_0 = [O_0, I_0, J_0, K_0]$ and $L_1 = [O_1, I_1, J_1, K_1]$, is a pose-valued function $L(t)$ satisfying the constraints $L(0) = L_0$ and $L(1) = L_1$. 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 (i.e. helical) 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** (i.e. piecewise

[†] e-mail: apowell@anim.dreamworks.com

[‡] now at DreamWorks Animation

[§] e-mail: jarek@cc.gatech.edu

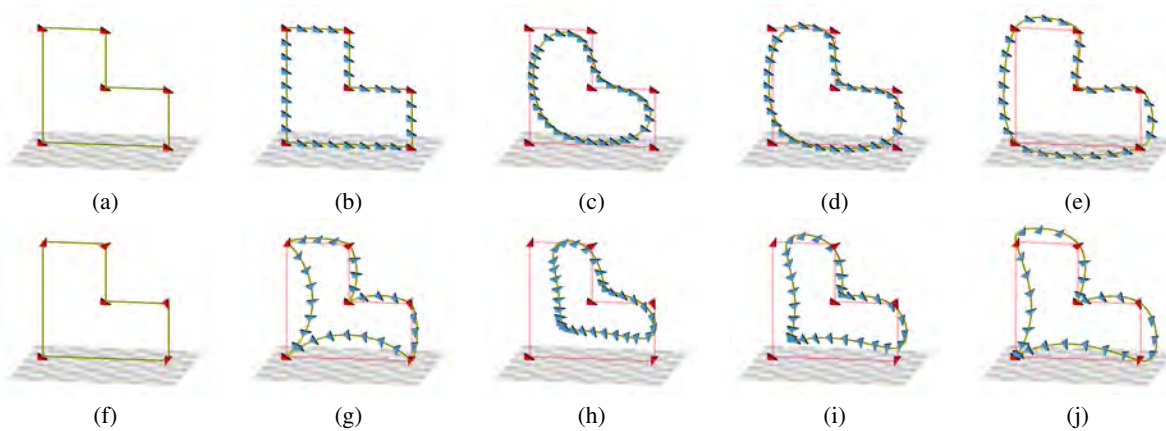


Figure 1: A cycle of parallel control poses defines an animation with sharp changes of velocity at control poses (a). The trajectory of a polygonal translation interpolating them (b) may be smoothed by a cubic B-spline (c), Jarek (d), or 4-point polygon subdivision (e). When four of the control poses 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 .

helical) motion. The term polyscrew is chosen to emphasize its relation with a polygon. Although continuous, 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 (introduced by Wallner and Pottman [WP06]) 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 13).

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

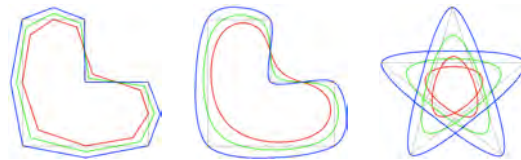


Figure 2: The original L-shaped control polygon (gray) is subdivided once (left) and then 5 times (center) by the B-spline (red), by the 4-point (blue), and by the Jarek (green) schemes. Corresponding subdivisions of a more complex polygon is shown (right).

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 [DGL87] 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. Note that the Catmull-Rom schemes [CR74], which convert the control polygon into a series of cubic Bezier arcs connected with tangent continuity, produce results similar to those produced

by the 4-point subdivision. They will not be explored here. The Jarek subdivision [Ros04] uses, at each step, for each old and new vertex, the average of the positions 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.

To prove the smoothness of the Jarek subdivision, we follow an outline provided by Scott Schaefer based on the analysis of the norms of Laurent polynomials [WW01, WD05]. For the three schemes discussed here, a sequence \mathbf{p}_i of vertices is mapped to a sequence \mathbf{q}_i as follows:

$$\begin{aligned} \text{B-SPLINE: } \quad \mathbf{q}_{2i} &= (2\mathbf{p}_{i-1} + 12\mathbf{p}_i + 2\mathbf{p}_{i+1})/16, \\ &\quad \mathbf{q}_{2i+1} = (8\mathbf{p}_i + 8\mathbf{p}_{i+1})/16 \\ \text{4-POINT: } \quad \mathbf{q}_{2i} &= (16\mathbf{p}_i)/16, \\ &\quad \mathbf{q}_{2i+1} = (-\mathbf{p}_{i-1} + 9\mathbf{p}_i + 9\mathbf{p}_{i+1} - \mathbf{p}_{i+2})/16 \\ \text{JAREK: } \quad \mathbf{q}_{2i} &= (2\mathbf{p}_{i-1} + 28\mathbf{p}_i + 2\mathbf{p}_{i+1})/32, \\ &\quad \mathbf{q}_{2i+1} = (-\mathbf{p}_{i-1} + 17\mathbf{p}_i + 17\mathbf{p}_{i+1} - \mathbf{p}_{i+2})/32 \end{aligned}$$

Note that the weights for the Jarek scheme are the averages of the corresponding B-spline and 4-point weights. We compute the coefficients of the Laurent polynomial for Jarek as:

$$\begin{aligned} S(z) &= (-1 + 2z + 17z^2 + 28z^3 + 17z^4 + 2z^5 - 1z^6)/32 \\ &= -(1+z)^4(1-6z+z^2)/32 \end{aligned}$$

To establish that Jarek subdivision is C^0 , we form $T(z) = S(z)/(1+z) = -(1+z)^3(1-6z+z^2)/32 = (-1+3z+14z^2+14z^3+3z^4-z^5)/32$. Then, we compute the sum of the absolute values of the even power entries and the sum of the absolute values of the odd power entries. The infinity norm, $\|T(z)\|_\infty = 9/16$, is the larger of these two sums. Since it is less than 1, Jarek subdivision is C^0 [DGL91].

To establish that it is C^1 , we check whether $D(z) = 2S(z)/(1+z)$ is continuous. Hence, we have $D(z) = -(1+z)^3(1-6z+z^2)/16 = (-1+3z+14z^2+14z^3+3z^4-z^5)/16 = (1+z)(-1+4z+10z^2+4z^3-z^4)/16$. The infinity norm of $D(z)/(1+z)$ is $\|D(z)\|_\infty = 3/4$. We conclude that the Jarek scheme is C^1 .

To establish that it is C^2 , we check whether $Q(z) = 4S(z)/(1+z)^2$ is continuous. Hence, we have $Q(z) = -(1+z)^2(1-6z+z^2)/8 = (-1+4z+10z^2+4z^3-z^4)/8 = (1+z)(-1+5z+5z^2-1z^3)/8$. The infinity norm of $Q(z)/(1+z)$ is $\|Q(z)\|_\infty = 3/4$. We conclude that the Jarek scheme is C^2 . This observation is illustrated in Figure 3.

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.

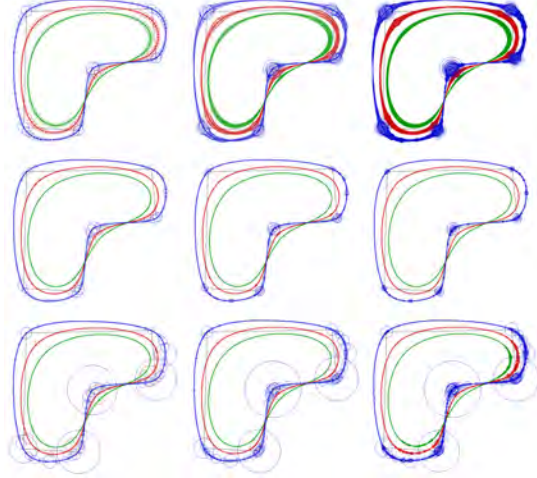


Figure 3: We show the results of successive subdivisions using the cubic B-spline (green), the Jarek (red), and the 4-point (blue) schemes. Let $V[j]$ denote the j^{th} vertex. $D_0[j] = V[j+1] - V[j]$. $n[j] = \|D_0[j]\|$. $D_i[j] = (D_{i-1}[j+1] - D_{i-1}[j])/n$. Radii of circles represent the scaled magnitudes of the corresponding D_i vectors after 4 (left column), 6 (center column), and 8 (right column) subdivision steps. Top row ("curvature"): $128D_1$, $512D_1$, and $2048D_1$. Middle row ("jerk"): $4096D_2$. Bottom row: $131072D_3$, $32768D_3$, and $8192D_3$. Notice that the peaks in curvature variation D_2 (middle row) decrease with levels of subdivision for B-spline and Jarek (which are C^2), but not for 4-points (which is C^1).

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 changes of orientation 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

The ScrewBender system combines these two ideas (Split&Tweak [Ros04] and polyscrews [RK01]) and defines subdivision rules for polyscrews that correspond to the cubic B-spline, Jarek, and 4-point subdivisions of polygons. It only requires the ability to compute and evaluate the screw that interpolates a pair of control poses.

We do so with the procedure $s(L_a, t, L_b)$, which returns the pose $L_{a,b}(t)$, obtained by moving L_a by a fraction t of the screw motion defined by L_a and L_b . In particular, $L_{a,b}(0) = L_a$ and $L_{a,b}(1) = L_b$. To compute $s(L_a, t, L_b)$, we compute the screw parameters S , r , d , and P , as discussed in Section 3, then translate L_a by tdS and rotate it by angle tr around the screw axis P with direction S .

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 4).

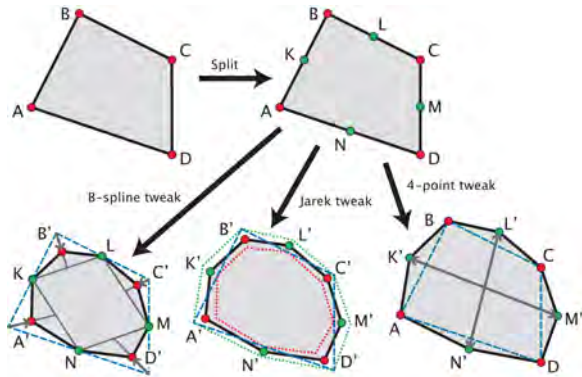


Figure 4: The polygon (top-left) is split by inserting a new vertex in the middle of each edge. Then, vertices are tweaked to complete the B-spline, Jarek, or 4-point subdivision step.

A variety of constructions may be used for the new vertices in the 4-point scheme. They are all equivalent for the

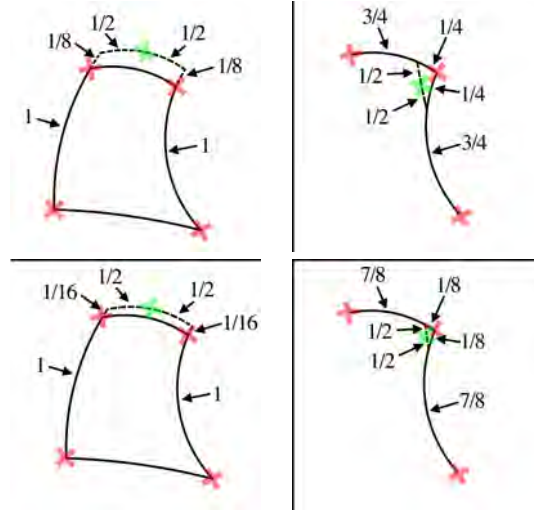


Figure 5: An example is used to demonstrate the Split&Tweak process for polyscrew subdivision. Top left: A new pose is inserted and tweaked in 4-point subdivision. Top right: After new poses are inserted in B-spline subdivision, an old pose is tweaked. The Jarek subdivision process involves tweaking the new vertices (bottom left) and tweaking the old (bottom right).

linear case (polygons or pure translation motion). However, they may produce different results when subdividing polyscrews. To illustrate these differences, we compare two of the more obvious constructions (Figure 6).

In the preferred formulation (A), shown in Figure 6 (left), at each step of the subdivision, and for each screw, we insert a new control pose $s(s(L_a, \frac{9}{8}, L_b), \frac{1}{2}, s(L_d, \frac{9}{8}, L_c))$ between control poses L_b and L_c in the subsequence of poses (L_a, L_b, L_c, L_d) . In formulation (B), we use $s(s(L_a, \frac{1}{2}, L_d), \frac{9}{8}, s(L_b, \frac{1}{2}, L_c))$. When every set of 4 consecutive control poses has similar orientation (for example when the screws between every pair have less than a 45 degree angle), formulations (A) and (B) produce very similar motions. In fact, the difference is hard to notice. However, as the angle of screw $L_{a,d}$ approaches 180 degrees, formulation (B) may produce unacceptable motions when the directions of the screw axes of $L_{a,d}$ and $L_{b,c}$ have opposite orientations (i.e. their dot product is negative), as shown in Figure 6. This flip corresponds to the elbow up/down flip often encountered when computing the inverse kinematic for robots or articulated bodies in animation. These situations may be detected and corrected by reversing the direction of axis $L_{a,d}$ and by replacing its angle r by $360 - r$ (as was done in [LPRS05]). However, since formulation (A) avoids the problem in situations where the motion is reasonably consistent and produces results similar to (B), we advocate using it.

Note that one can produce a different smooth interpolating

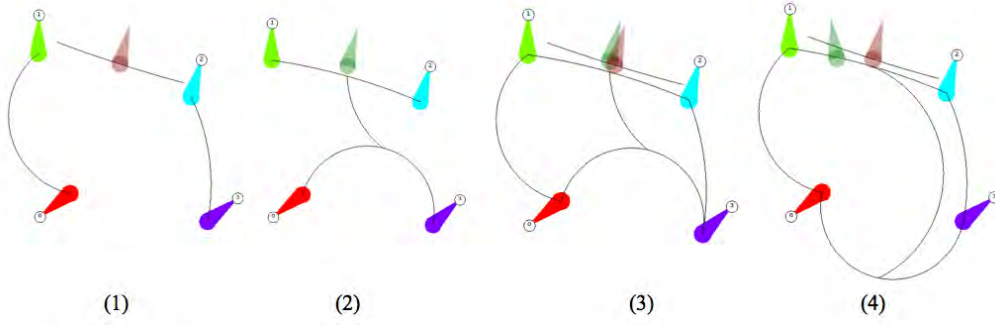


Figure 6: (1) The proposed 4-point construction, formulation (A), inserts a new pose in the middle of a screw between $s(L_a, \frac{9}{8}, L_b)$ and $s(L_d, \frac{9}{8}, L_c)$. (2) The alternate construction, formulation (B), computes $L_n = s(L_a, \frac{1}{2}, L_d)$ and $L_m = s(L_b, \frac{1}{2}, L_c)$ then computes the new pose as $s(L_m, \frac{9}{8}, L_n)$. (3) In general, the two approaches produce similar results. (4) A slight rotation of control pose number 3, when the angle of screw $L_{a,d}$ approaches 180 degrees, may produce a flip in the direction of the axis of L_n when formulation (B) is used. Note that formulation (A) is resilient to this change.

piecewise helical motion [Sho85] that mimics the Catmull-Rom spline constructions [CR74], which is slightly more complex than the 4-point rule.

For the B-spline scheme (Figure 5, top right), we insert a new control pose $L_{a,b}(\frac{1}{2})$ between each pair of consecutive poses L_a and L_b to 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 of old poses (L_a, L_b, L_c) by $s(s(L_b, \frac{1}{4}, L_a), \frac{1}{2}, s(L_b, \frac{1}{4}, L_c))$. Although other constructions may appear more natural, for example $s(L_b, \frac{1}{4}, s(L_a, \frac{1}{2}, L_c))$, the one proposed above does an excellent job at avoiding the elbow flip problem mentioned earlier.

For the Jarek scheme (Figure 5, bottom), we compute (but do not yet 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 of old poses (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. Again, we have found that this construction avoids the elbow flip problems encountered when using constructions that compute intermediate poses farther from the old ones.

5. Results and Applications

In ScrewBender, the smoothly subdivided L-shaped polyscrew of Figure 1(j) may be edited by dragging or rotating a graphically selected control pose (Figure 13). 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 of the entire motion. Our non-optimized implementation

can compute more than 110,000 $s(L_a, t, L_b)$ 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 7.



Figure 7: 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 8a. 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 8b).

6. Smoothness

The smoothness benefits of the 4-point polyscrew subdivision are illustrated in Figure 9, where we show a stroboscopic superposition of numerous instances of a triangle animated by a polyscrew motion. The original polyscrew motion is clearly not C^1 , while the subdivided one appears to be C^1 .

Sometimes, the degree of smoothness of a motion may

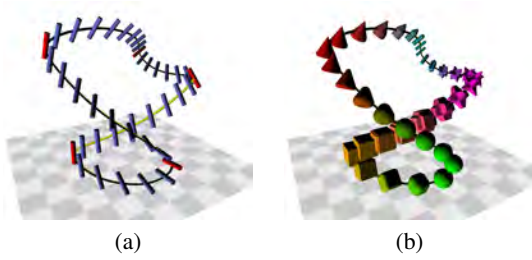


Figure 8: A stroboscopic rendering of the motion of a cylinder (a). In (b), a different shape is associated with each control pose of the animation. As the object moves along the 4-point subdivided polyscrew, its shape morphs between a cone, a star, a block, a cylinder, and a sphere.

be derived by analyzing the fairness of the curve that represents it in the Lie group $SE(3)$ [PR97]. Here we examine the smoothness of subdivided polyscrew motions 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 10. 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. We also conjecture that the Jarek subdivision yields at least C^1 trajectories.

Several authors [WD05, WP06, WYG06] have proven the smoothness of nonlinear subdivision schemes viewed as perturbations of linear ones by showing that they have similar properties with regards to C^1 and C^2 smoothness as the linear schemes they are based upon when specific proximity conditions are fulfilled. In particular Wallner and Pottmann have used this approach to prove that perturbing the linear 4-point and cubic B-spline schemes to their helical counterparts satisfies these proximity conditions. We conjecture that their approach could be extended to prove that the Jarek subdivision of polyscrews is C^1 and possibly even C^2 .

7. Contributions in the Context of Prior Art

Shoemake [Sho85, SD92] used spherical linear interpolation of quaternions (SLERP) to define interpolations of shape orientations. 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.

Möller and Hughes [MH99] proposed an efficient computation of the matrix that rotates one vector to another, but without consideration of smoothness from one rotation to the

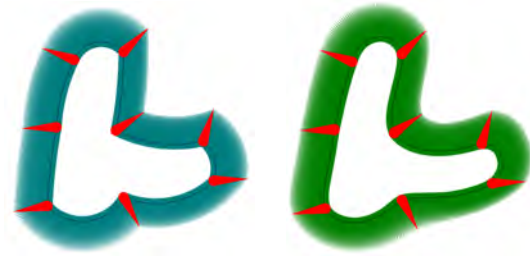


Figure 9: A planar motion of a triangle is defined by 8 control poses (red). Consider the trajectory T of the tip of this triangle, as shown using the superposition of a large number of frames of the animation. On the left, the frames are produced by interpolating each pair of consecutive control poses by a screw. Because the motion is planar, each screw is either a pure rotation or a translation. Notice that T is piecewise circular, but exhibits tangent discontinuities at the control poses. On the right, this original polyscrew motion has been subdivided through a four-point subdivision scheme. Notice that T appears to be C^1 continuous.

next. Grassia [Gra98] promoted the use of 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 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 8b). 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 11a). 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

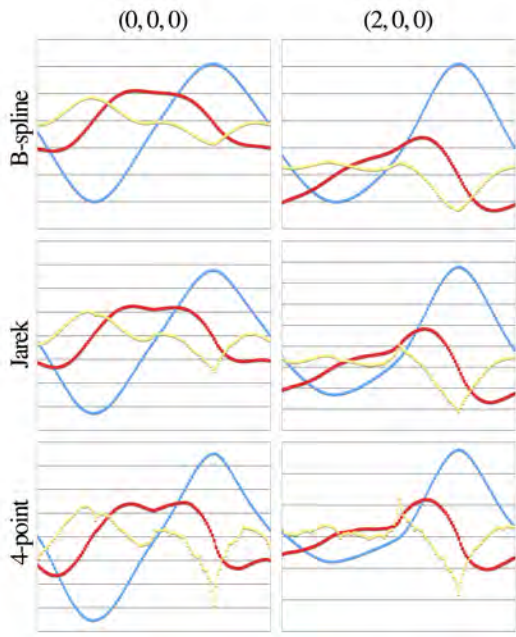


Figure 10: Plots of the x-coordinate of the points (0,0,0) (left column) and (2,0,0) (right column) on the moving object (blue), of its first derivative (red), and of its second derivative (gold) as a function of time for the B-spline, Jarek, and 4-point polyscrew subdivisions.

here, does not preserve the initial trajectory (Figure 11b) and produces a somewhat wobbly motion.

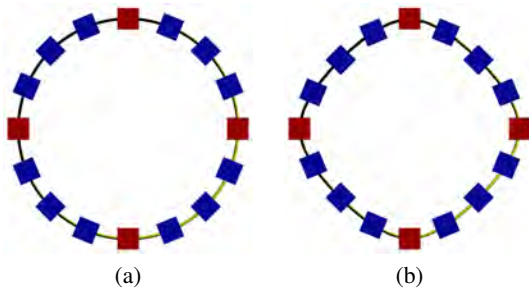


Figure 11: When subdivided with any of the polyscrew subdivisions presented here, a circular motion defined by four control poses (shown in red) retains its initial trajectory (a). A decoupled smoothing results in a less natural subdivided trajectory (b).

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 $s(A, \frac{1}{2}, B)$ 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] have proposed subdivision schemes for piecewise helical motions and show that the C^1 and C^2 smoothness of linear subdivision methods extend to helical subdivision. Although developed independently and based on a different construction, their schemes are variations of our Split&Tweak B-spline and 4-point subdivisions of polyscrews. Hence, our contribution is the extension of these piecewise helical motion subdivision schemes to the Jarek subdivision and a simple construction leading to their realtime subdivision and animation, which makes them particularly attractive for direct manipulation.

8. 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.

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.

9. Acknowledgments

The authors wish to thank the reviewers for suggesting a simple proof that the Jarek subdivision is at least C^1 , Malcolm Sabin and Scott Schaefer for suggesting that it is actually C^2 , and Scott Schaefer for suggesting a proof of that fact.

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.
- [CR74] CATMULL E., ROM R.: A class of local interpolating splines. *Computer Aided Geometric Design* (1974), 317–326.
- [DGL87] DYN N., GREGORY J. A., LEVIN D.: A four-point interpolatory subdivision scheme for curve design. *Computer Aided Geom. Design* 4 (1987), 257–268.
- [DGL91] DYN N., GREGORY J. A., LEVIN D.: Analysis of uniform binary subdivision schemes for curve design. *Constructive Approximation* 7 (1991), 127–147.
- [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.
- [LPRS05] LLAMAS I., POWELL A., ROSSIGNAC J., SHAW C.: Bender: A virtual ribbon for deforming 3d shapes in biomedical and styling applications. *ACM Symposium on Solid and Physical Modeling (SPM)* (June 2005), 89–99.
- [LR80] LANE J. M., RIESENFELD R. F.: A theoretical development for the computer generation and display of piecewise polynomial surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 2, 1 (1980), 35–46.
- [MH99] MÖLLER T., HUGHES J. F.: Efficiently building a matrix to rotate one vector to another. *J. Graph. Tools* 4, 4 (1999), 1–4.
- [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*, Farin G., Hoschek J., 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.
- [WD05] WALLNER J., DYN N.: Convergence and c^1 analysis of subdivision schemes on manifolds by proximity. *Computer Aided Design* 22, 7 (2005), 593–622.
- [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.
- [WW01] WARREN J., WEIMER H.: Subdivision methods for geometric design: A constructive approach. *The Morgan Kaufmann Series in Computer Graphics* (2001).
- [WYG06] WALLNER J., YAZDANI E. N., GROHS P.: Smoothness properties of lie group subdivision schemes. *Technical Report 165, Geometry Preprint Series* (November 2006).

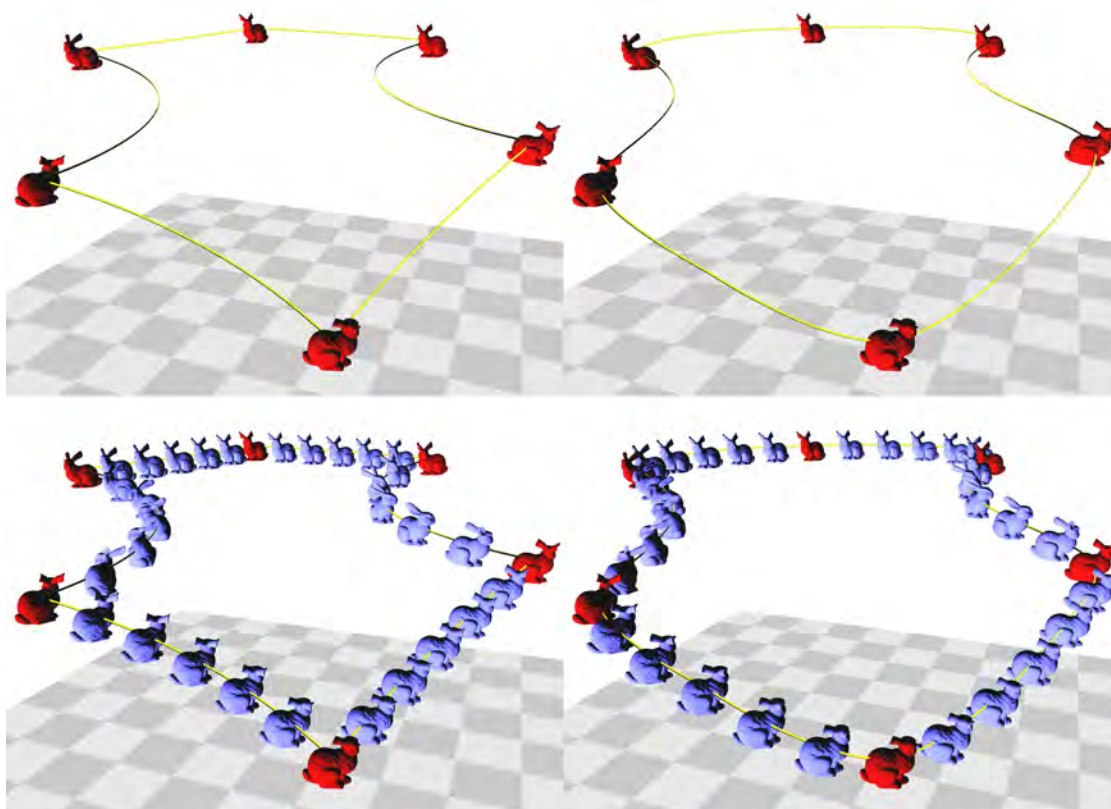


Figure 12: A cyclic animation of a non-planar motion of bunny is defined with 6 control poses (shown in red). Top left: The path for an animation composed of single screws that each interpolate two consecutive control poses. Notice the sharp direction discontinuities at the control poses. Bottom left: 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.

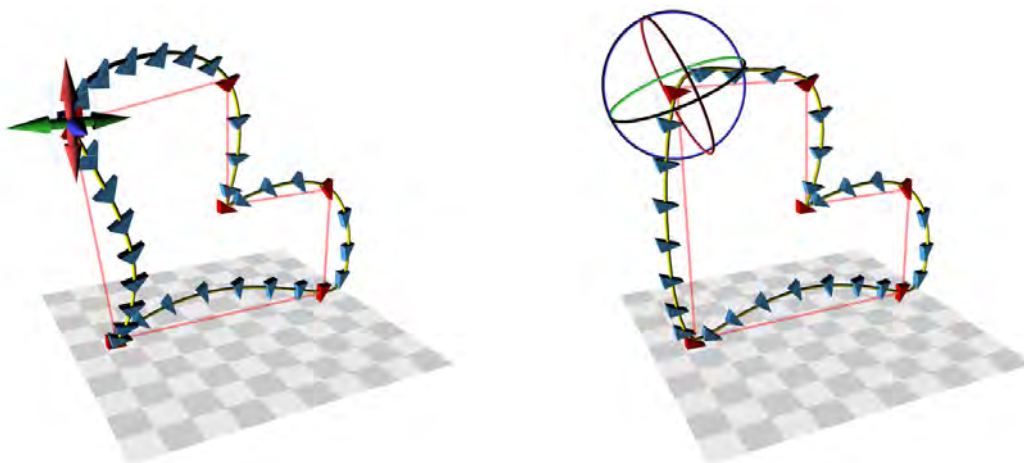


Figure 13: The stroboscopic rendering of the subdivided polyscrew in Figure 1 is updated in realtime as the user translates (left) or rotates (right) the top-left control pose of the model. Note that the user has local control over the smooth motion.