Parameterization and Applications of Catmull-Rom Curves

Cem Yuksel, Scott Schaefer, John Keyser

3112 Texas A&M University
College Station, TX 77843-3112

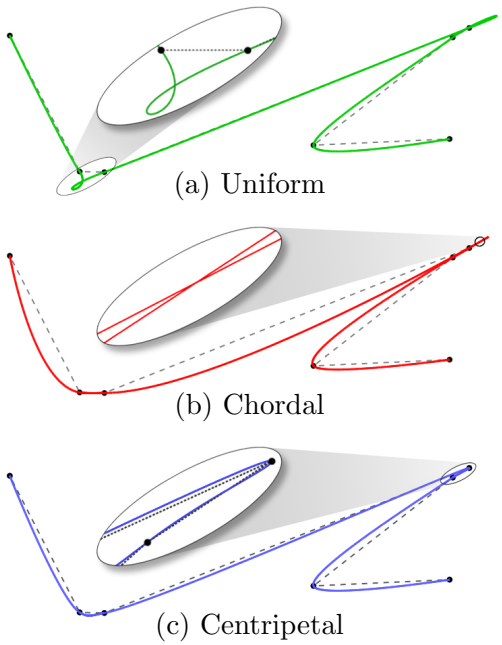
Abstract

The behavior of Catmull-Rom curves heavily depends on the choice of parameter values at the control points. We analyze a class of parameterizations ranging from uniform to chordal parameterization and show that, within this class, curves with centripetal parameterization contain properties that no other curves in this family possess. Researchers have previously indicated that centripetal parameterization produces visually favorable curves compared to uniform and chordal parameterizations. However, the mathematical reasons behind this behavior have been ambiguous. In this paper we prove that, for cubic Catmull-Rom curves, centripetal parameterization is the only parameterization in this family that guarantees that the curves do not form cusps or self-intersections within curve segments. Furthermore, we provide a formulation that bounds the distance of the curve to the control polygon and explain how globally intersection-free Catmull-Rom curves can be generated using these properties. Finally, we discuss two example applications of Catmull-Rom curves and show how the choice of parameterization makes a significant difference in each of these applications.

Keywords: Catmull-Rom splines, parameterization, chordal parameterization, centripetal parameterization, uniform parameterization, animation curves, path curves

1. Introduction

Catmull-Rom curves are widely used in graphics for a variety of applications ranging from modeling to animation. These parametric curves have three important properties that make them so popular. First, the curves



(a) Uniform

(b) Chordal

(c) Centripetal

Figure 1: Cubic Catmull-Rom curves with (a) uniform, (b) chordal, and (c) centripetal parameterization. While uniform and chordal parameterizations can produce self-intersections, centripetal parameterization is the only one that guarantees no self-intersections within curve segments.

are smooth and interpolate their control points, which gives the user direct control over various points on the curve. Second, the curves have local support, so that each control point only affects a small neighborhood on the curve. Finally, Catmull-Rom curves have an explicit piecewise polynomial representation, allowing them to be easily be converted to other bases and manipulated computationally.

Perhaps the most popular parameterization of Catmull-Rom curves is a uniform parameterization (i.e. the control points are equally spaced in parametric space). However, this choice of parameterization does not reflect the Euclidean distance between control points well. For curves with different length segments, this parameterization can lead to artifacts such as cusps and self-intersections, which occur frequently (Figure 1a). Moreover, the distance of the curve from the control polygon can be unbounded, which makes these curves difficult to control in practice.

An alternative is to automatically create the parameterization of the curve from its geometric embedding in Euclidean space. Doing so gives rise to other

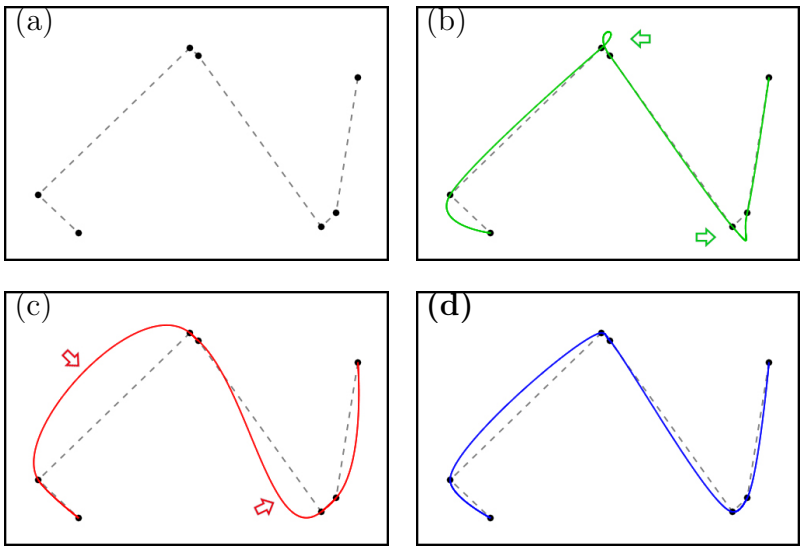


Figure 2: Catmull-Rom curves generated using the same control polygon (a) with different parameterizations. Uniform parameterization (b) overshoots and often generates cusps and intersections within short curve segments, while chord-length parameterization (c) exhibits similar behavior for longer curve segments. Centripetal parameterization (d) is the only one that guarantees no intersections within curve segments.

known curve parameterizations such as chordal and centripetal parameterizations (Figure 1bc). However, like uniform parameterization, most parameterization choices still produce the same artifacts observed with uniform parameterization (cusps, self-intersections, etc...).

Researchers have previously compared uniform, chordal, and centripetal parameterizations for various curves [1, 2, 3, 4] and observed that, among these three parameterization choices, centripetal parameterization produces visually favorable curves. Yet, the reasoning behind this preference has been limited to informal explanations based on intuition, rather than a more formal mathematical explanation. Floater [5] does provide some evidence that, among these three parameterizations, centripetal parameterization produces curves closer to the control polygon for cubic splines than uniform or chordal parameterization. However, centripetal parameterization can be considered as just one choice within an infinite family of parameterization choices between uniform and chordal. Therefore, there may exist some other parameterization that would produce even more favorable results than centripetal parameterization.

In this paper, we concentrate on cubic Catmull-Rom curves and analyze the full class of parameterizations ranging from uniform to chordal parameterization, such that the parameterization is a function of the length between two consecutive control points. We show that centripetal parameterization, which is at the center of this class, inherits some important properties that no other parameterization in this class possesses for these curves. Following a brief overview of Catmull-Rom curves in Section 2, in Section 3 we mathematically prove that centripetal parameterization of Catmull-Rom curves guarantees that the curve segments cannot form cusps or local self-intersections, while such undesired features can be formed with all other possible parameterizations within this class. Furthermore, we provide a formulation that bounds the distance between the control polygon and the actual curve in Section 4. Based on these two properties we derive rules to achieve globally intersection-free Catmull-Rom curves in Section 5. In Section 6, we provide a discussion of our results and observations. Finally, we explain the effect of parameterization in two applications of Catmull-Rom curves in Section 7, before we conclude in Section 8.

2. Background

Catmull-Rom curves were first described in [6] as a method for generating interpolatory curves with local support by combining Lagrange interpolation and B-spline basis functions. Barry and Goldman [7] exploited this relationship to show how to construct non-uniform Catmull-Rom curves by factorizing the computation into a pyramid. Let $\mathbf{P}_i \in \mathbb{R}^m$ be the control points of a Catmull-Rom curve and each control point be associated with the parametric value s_i. A C^k Catmull-Rom curve is composed of polynomial segments of degree $2k+1$ between consecutive control points. These polynomial pieces are only affected by a local set of control points. The polynomial piece of the curve between s_i and s_{i+1} is influenced by control points $\mathbf{P}_{i-k}$ through $\mathbf{P}_{i+1+k}$. Furthermore, the curve is interpolatory (i.e. at s_i and s_{i+1}, the curve evaluates to $\mathbf{P}_i$ and $\mathbf{P}_{i+1}$ respectively).

We concentrate on C^1 cubic Catmull-Rom curves as they are the simplest and most popular form of these curves. Figure 3 shows Barry and Goldman's pyramid algorithm for cubic Catmull-Rom curves that builds the polynomial $C_{12}(s)$ for the curve segment between parameter values s_1 and s_2. This pyramid is composed of triangles with two points at the base and arrows with coefficients leading to its apex. This notation should be interpreted as

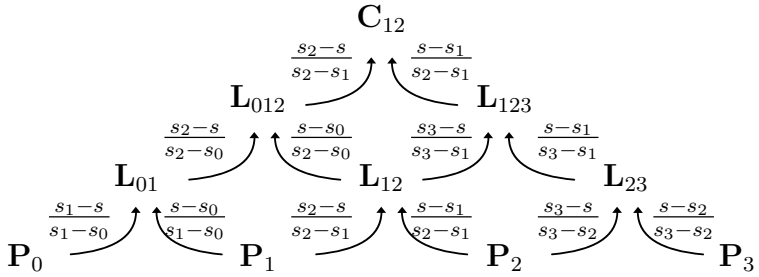


Figure 3: Cubic Catmull-Rom curve formulation.

multiplying each point at the base of the triangle by the coefficient on the arrow and summing the result. From this diagram, it is easy to see that C^1 Catmull-Rom curves are cubic polynomials as there are 3 levels in this pyramid and each adds a single, linear factor.

Notice that Barry and Goldman's description of the Catmull-Rom curve is non-uniform and allows for arbitrary s_i values. The choice of these s_i is what we refer to as the parameterization of the Catmull-Rom curve. The behavior of these curves depends significantly on the parameterization as shown in Figure 2. Various parameterization methods have been developed previously [4, 8, 9] and we analyze a class of parameterizations described by [4] ranging from uniform to chordal parameterization where we define the parameter values as

$$s_{i+1} = |\mathbf{P}_{i+1} - \mathbf{P}_i|^\alpha + s_i \ , \tag{1}$$

and $s_0 = 0$, where $0 \leq \alpha \leq 1$. Note that when $\alpha = 0$, the parameterization is uniform, and when $\alpha = 1$, the parameterization becomes the chordal parameterization. Similarly, $\alpha = \frac{1}{2}$ corresponds to centripetal parameterization.

3. Cusps and Self-Intersections

Cusps and self-intersections are very common with Catmull-Rom curves for most parameterization choices. In fact, as we will show here, the only parameterization choice that guarantees no cusps and self-intersections within curve segments is centripetal parameterization.

To determine if a curve segment of the Catmull-Rom curve has a self-intersection, we will convert the polynomial to Bézier form. Let $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ be four consecutive control points of the Catmull-Rom curve with parameter values $0, d_1^\alpha, d_2^\alpha + d_1^\alpha, d_3^\alpha + d_2^\alpha + d_1^\alpha$, where $d_i = |\mathbf{P}_i - \mathbf{P}_{i-1}|$ as shown in Figure 4. The control points of the cubic Bézier curve $\mathbf{B}_j$

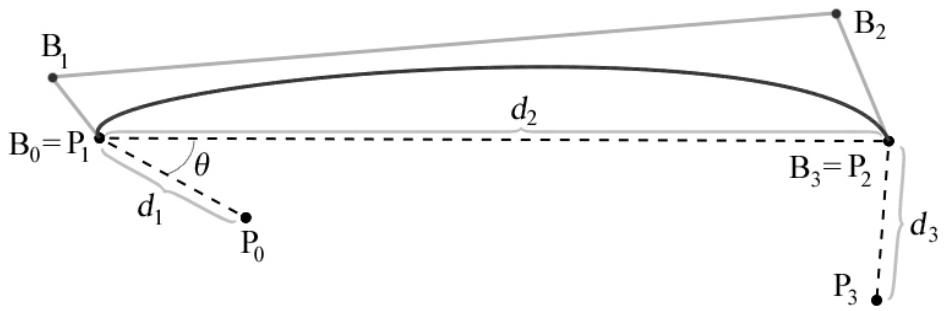


Figure 4: Control points $\mathbf{B}_j$ of the cubic Bézier curve constructed from cubic Catmull-Rom curve segment with control points $\mathbf{P}_i$.

($j \in \{0, 1, 2, 3\}$) representing this polynomial between d_1^α and $d_2^\alpha + d_1^\alpha$, reparameterized to lie in the range $[0, 1]$ are then

$$\begin{aligned}
\mathbf{B}_0 &= \mathbf{P}_1 \\
\mathbf{B}_1 &= \frac{d_1^{2\alpha}\mathbf{P}_2 - d_2^{2\alpha}\mathbf{P}_0 + (2d_1^{2\alpha} + 3d_1^\alpha d_2^\alpha + d_2^{2\alpha})\mathbf{P}_1}{3d_1^\alpha(d_1^\alpha + d_2^\alpha)} \\
\mathbf{B}_2 &= \frac{d_3^{2\alpha}\mathbf{P}_1 - d_2^{2\alpha}\mathbf{P}_3 + (2d_3^{2\alpha} + 3d_3^\alpha d_2^\alpha + d_2^{2\alpha})\mathbf{P}_2}{3d_3^\alpha(d_3^\alpha + d_2^\alpha)} \\
\mathbf{B}_3 &= \mathbf{P}_2.
\end{aligned} \tag{2}$$

A smooth curve will not have cusps or self-intersect on the parameter range $[0, 1]$ if there exists a line such that the curve projected onto this line has derivative greater than zero over that interval [10]. Our choice of projection will be the line connecting $\mathbf{B}_0$ and $\mathbf{B}_3$ as this is the only choice that is applicable to curves of all dimensions. Note that this condition with our choice of projection is both a necessary and sufficient condition for 1D curves (i.e. when the $\mathbf{B}_j$ are co-linear), and is a necessary but not sufficient condition for higher dimensions.

We will first show that parameterizations other than centripetal can produce cusps and self intersections by analyzing the derivative of the curve at the endpoints. We will then show that for centripetal parameterization, it is not possible to produce cusps or self intersections.

Theorem 1. *For parameterizations of cubic Catmull-Rom curves other than centripetal, the projected derivative may be negative at the end-points.*

Proof. Given that our curve is a cubic and that the axis we have chosen connects the two end-points of the Bézier curve, we need only consider the vector $\mathbf{B}_1 - \mathbf{B}_0$ in relationship to our chosen axis ($\mathbf{B}_2 - \mathbf{B}_3$ follows through

symmetry). Hence if $(\mathbf{B}_1 - \mathbf{B}_0) \cdot (\mathbf{B}_3 - \mathbf{B}_0) < 0$, then the projected derivative will begin negative (the direction of the derivative of a Bézier at its end-points is given by the vector from the end-point to the adjacent control point). Expanding this expression using Equation 2 and the property that $d_i = |\mathbf{P}_i - \mathbf{P}_{i-1}|$, yields

$$\frac{d_1^{2\alpha} d_2^2 - d_2^{2\alpha+1} d_1 \cos(\theta)}{3 d_1^{\alpha}(d_1^{\alpha} + d_2^{\alpha})} < 0 \tag{3}$$

where θ is the angle between $\mathbf{P}_0 - \mathbf{P}_1$ and $\mathbf{P}_2 - \mathbf{P}_1$ as shown in Figure 4. The left-hand side of this inequality achieves its minimum when $\cos(\theta) = 1$ and the expression simplifies to

$$d_1^{2\alpha} d_2 < d_1 d_2^{2\alpha}.$$

When $\alpha < \frac{1}{2}$, this expression is satisfied when $d_2 < d_1$. When $\alpha > \frac{1}{2}$, this expression is satisfied when $d_1 < d_2$. The only value of α that cannot meet this inequality is $\alpha = \frac{1}{2}$. Hence, the centripetal parameterization is the only parameterization for which the projected derivative at the endpoint is always non-negative. □

Since the derivative must be positive somewhere for the curve to reach $\mathbf{B}_3$ and the derivative is continuous, a negative derivative at the endpoint implies that a cusp or self intersection can be created. Thus, centripetal parameterization offers the only possibility for avoiding cusps and local self intersections.

This test, however, is not sufficient to show that centripetal parameterization cannot produce cusps within a single polynomial. We will show this property in two stages, first by proving a general property regarding cusp formation, and then by showing that centripetal parameterization meets the requirements of that property.

Theorem 2. *A cubic Bézier curve whose interior control points project to be within the open interval defined by the end-points of the Bézier curve cannot have a cusp or self-intersection.*

Proof. Since Bézier curves are affinely invariant, we can assume without loss of generality that $\mathbf{B}_0$ is at the origin and $\mathbf{B}_3$ is on the x-axis at $x = 1$. The control points for the projected curve will be univariate values as well and the control points for the projected Bézier curve are then $(0, x_1, x_2, 1)$ where

x_1, x_2 are the x components of $\mathbf{B}_1$ and $\mathbf{B}_2$. Our goal is to show that this projected curve cannot have a zero derivative over this interval.

To this end, we construct the control points of the derivative of this curve, which is a quadratic Bézier curve with control points $(3x_1, 3(x_2-x_1), 3-3x_2)$. Our assumption in the theorem states that $0 < x_1 < 1$ and $0 < x_2 < 1$. Therefore, there are two cases to consider: $x_1 \leq x_2$ and $x_1 > x_2$.

If $x_1 \leq x_2$, then the control points of the derivative curve are all greater than or equal to zero and, by the convex hull property of Bézier curves, the derivative is greater than zero.

If $x_1 > x_2$, then we can solve for the minimum of this quadratic Bézier polynomial, which is

$$\frac{3(x_1(1-x_1) + x_2(x_1-x_2))}{1+3(x_1-x_2)}.$$

Notice that the denominator is always positive, since $x_1 > x_2$. Furthermore, $x_1(1-x_1) > 0$ because $0 < x_1 < 1$, and $x_2(x_1-x_2) > 0$ since $0 < x_2$ and $x_1 > x_2$. Therefore, the numerator is always positive as well and the derivative is always greater than zero. □

Theorem 2, assumes that the projection of the interior control points lies in the range $(0, 1)$. We must show this is the case for centripetal parameterization.

Theorem 3. *For centripetal parameterization of cubic Catmull-Rom curves, the interior control points of a cubic Bézier curve may not project beyond the outer control points.*

Proof. We can again consider only the case of $\mathbf{B}_1$ (since $\mathbf{B}_2$ follows by symmetry). Consider the projected magnitude of $\mathbf{B}_1 - \mathbf{B}_0$ onto the edge defined by the end-points of the Bézier curve.

$$\frac{(\mathbf{B}_1 - \mathbf{B}_0) \cdot (\mathbf{B}_3 - \mathbf{B}_0)}{|\mathbf{B}_3 - \mathbf{B}_0|^2}.$$

If $\mathbf{B}_1$ projects onto the open interval defined by $\mathbf{B}_0$ and $\mathbf{B}_3$, then this quantity must be within the range $(0, 1)$. By Theorem 1, the numerator is non-negative and, hence, this quantity is greater than or equal to 0. The case when this quantity is equal to 0 corresponds to $\mathbf{B}_1 = \mathbf{B}_0$, which can indeed happen. However, this boundary case does not indicate a cusp as the derivative is

only zero exactly at the end-point. Therefore, to apply Theorem 2, we only must show that this quantity cannot satisfy

$$\frac{(\mathbf{B}_1 - \mathbf{B}_0) \cdot (\mathbf{B}_3 - \mathbf{B}_0)}{|\mathbf{B}_3 - \mathbf{B}_0|^2} \geq 1.$$

Using Equation 2 and letting $r = d_1/d_2$ be the ratio between the lengths of consecutive segments of the control polygon, this expression simplifies to

$$\frac{r^\alpha - r^{1-\alpha}\cos(\theta)}{3(1 + r^\alpha)} \geq 1. \tag{4}$$

This expression will be maximal when $\cos(\theta) = -1$. Using this substitution and rewriting the expression yields

$$r^{1-\alpha} \geq 3 + 2r^\alpha.$$

For $\frac{1}{2} \leq \alpha \leq 1$ this expression is obviously false. □

Thus, Theorem 2 guarantees that centripetal parameterization cannot produce cusps or self-intersections. Theorem 1 shows that this is the only parameterization of cubic Catmull-Rom curves with that guarantee.

4. Distance Bound

A commonly desired property in all of geometric modeling is that the control structure should provide some intuition about the shape being modeled. One typical way that this is expressed is that a curve should behave "similarly" to its control polygon. Other researchers [5] have also noted that a good interpolatory curve is one that does not deviate far from its control polygon. Thus, we would like to have a way to measure the possible deviation of a curve from its control polygon.

Consider the curve segment from $\mathbf{P}_1$ to $\mathbf{P}_2$ with Bézier points given by Equation 2. We will bound the distance of this curve to the line segment containing its end-points. To do so, we first bound the distance of the curve to the infinite line containing its end-points as a lower bound to the distance to the line segment itself.

To bound this distance to the infinite line, we first bound the distance of $\mathbf{B}_1$ and $\mathbf{B}_2$ to this line. Again, via symmetry, we only need to consider $\mathbf{B}_1$'s

distance to the infinite line

$$\begin{aligned} h_1 &= \sqrt{|\mathbf{B}_1 - \mathbf{B}_0|^2 - \left((\mathbf{B}_1 - \mathbf{B}_0) \cdot \frac{(\mathbf{B}_3 - \mathbf{B}_0)}{|\mathbf{B}_3 - \mathbf{B}_0|}\right)^2} \\ &= \frac{d_2^{2\alpha} d_1 \left|\sin(\theta)\right|}{3(d_1^\alpha + d_2^\alpha) d_1^\alpha} \end{aligned}$$

Substituting $r = d_1/d_2$ yields

$$h_1 = d_2 \frac{r^{1-\alpha} \left|\sin(\theta)\right|}{3(1 + r^\alpha)}. \tag{5}$$

Furthermore, the distance of a cubic Bézier curve to the infinite line containing its end-points is bounded by $\frac{3}{4}$ the distance of its control points to that line. Using this fact and the property that $|\sin(\theta)| \leq 1$, we can bound the distance h of any point on the curve to the infinite line by

$$h \leq d_2 \frac{r^{1-\alpha}}{4(1 + r^\alpha)}. \tag{6}$$

Notice that, for $\alpha < \frac{1}{2}$, this distance is potentially unbounded for arbitrary r. That is, for such parameterizations, we cannot bound the distance of the curve from the control polygon. However, for $\alpha \geq \frac{1}{2}$, this distance will be bounded solely as a fraction of the length of the edge (independent of r). For example, for both centripetal parameterization ($\alpha = \frac{1}{2}$) and chordal parameterization ($\alpha = 1$) the distance of the curve segment to the infinite line contained by its end-points is no more than $\frac{1}{4}$ times the length of the edge. The minimal bound (independent of r) is achieved when $\alpha = \frac{2}{3}$ where the ratio is $\frac{1}{8}$ times the length of the edge.

While centripetal and chordal parameterizations have similar bounds to the infinite line segments, they behave much differently in practice. For $\frac{1}{2} \leq \alpha < \frac{2}{3}$, the maximal distance ratio is achieved for $r > 1$ meaning that the line segment we are bounding distance to is smaller than its adjacent line segments (i.e. d_2 is relatively small). For $\frac{2}{3} < \alpha \leq 1$, the maximal distance ratio is achieved for $r < 1$, which is the case in which the line segment we are bounding distance to is large in comparison to the adjacent line segments (i.e. d_2 is relatively large). Since the bound in Equation 6 is related to the length of the line segment, d_2, the curve using chordal parameterization will appear farther away from the control polygon than the curve with centripetal

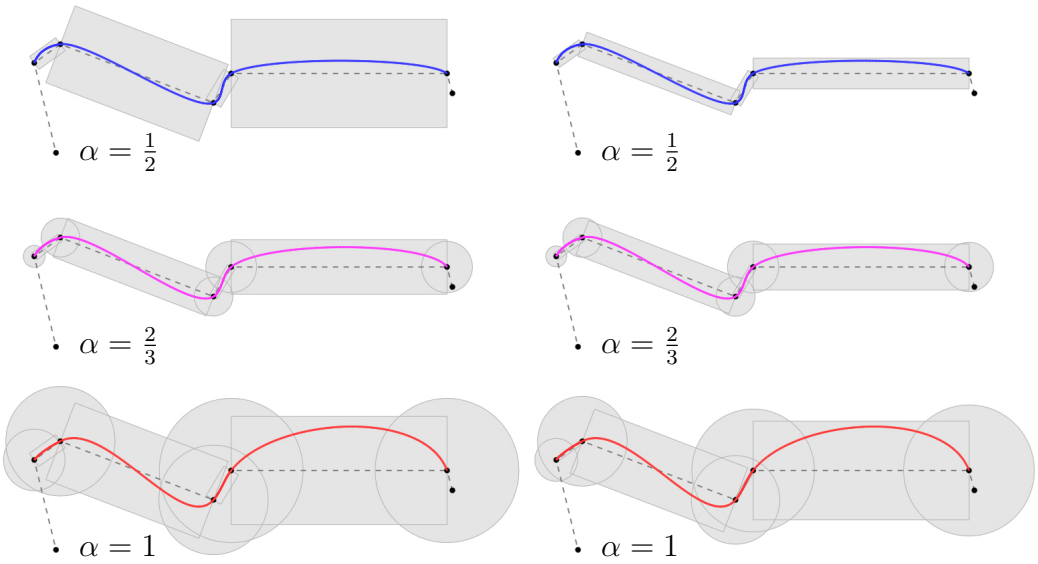


Figure 5: Bounding volumes for cubic Catmull-Rom curves with different α values. Bounds on the right column are computed using the length of the corresponding segment only, so that they represent maximum possible bound for the segment. Bounds on the left column also uses the lengths of neighboring segments using equations 6 and 8; therefore, they are more tight. Note that centripetal parameterization ($\alpha = \frac{1}{2}$) does not need circular bounds at the control points, because the curve is always confined in the boxes aligned with the edges of the control polygon.

parameterization, in absolute distance. In fact, for $\frac{1}{2} \leq \alpha \leq \frac{2}{3}$, the limit curve will never be farther than $\frac{1}{8}$ the length of the longest line segment in the control polygon and will typically be smaller. This effect can be seen in Figure 2, where the distance of the chordal curve is much further from the control polygon than the centripetal curve.

However, simply bounding the distance of the curve segment of a Catmull-Rom curve to the infinite line containing its end-points is not sufficient to bound the distance of the curve to the line *s*egment of the control polygon. When the interior Bézier points project outside of the line segment defined by $\mathbf{B}_0$ and $\mathbf{B}_3$, we must consider the distance of the control point $\mathbf{B}_1$ to its closest end-point. Notice that, by the discussion in Section 3, $\frac{1}{2} \leq \alpha \leq 1$ implied that $\mathbf{B}_1$ will never project outside the interval on the side of its opposite control point $\mathbf{B}_3$. Therefore, we only need to consider the length of the edge $\mathbf{B}_1 - \mathbf{B}_0$ to bound the distance of the curve to the end-point of the line segment.

We start by computing the angle θ at which the vector $\mathbf{B}_1 - \mathbf{B}_0$ becomes perpendicular to $\mathbf{B}_3 - \mathbf{B}_0$. This is the point at which we must start using the distance l_1 to the end-point $\mathbf{B}_0$ rather than the infinite line to bound the distance to the line segment. From the expression in Equation 3, we can find that

$$\cos(\theta) > r^{2\alpha-1} \, . \tag{7}$$

If we compute the length squared of the edge $\mathbf{B}_1 - \mathbf{B}_0$, we obtain

$$l_1^2 = |\mathbf{B}_1 - \mathbf{B}_0|^2 = \frac{d_2^2(r^2 + r^{4\alpha} - 2\cos(\theta)r^{1+2\alpha})}{9r^{2\alpha}(1+r^\alpha)^2}.$$

Notice that this equation depends on θ and is larger as $\cos(\theta)$ decreases. Combining this equation with the inequality constraints on $\cos(\theta)$ from Equation 7 and taking the square root of the expression, we can bound the maximal distance to the end-point as

$$l_1 \leq \frac{d_2\sqrt{r^2 - r^{4\alpha}}}{3r^\alpha(1+r^\alpha)}. \tag{8}$$

This expression is identically 0 for $\alpha = \frac{1}{2}$ meaning that a Catmull-Rom curve with centripetal parameterization can be bounded solely using bounding boxes extruded in the perpendicular direction of its line segments. However, for $\alpha = 1$, this bound can be as high as $\frac{1}{3}$ the length of the line segment. Therefore, for most curves we need not only bounding boxes around line segments but spheres around vertices to bound the curve completely. Figure 5 shows a 2D example of such bounding volumes. The figure demonstrates both the local bounds considering only the length of the line segment, as well as tighter bounds achieved by a (less local) evaluation of the lengths of adjacent segments using equations 6 and 8.

5. Intersection-free Curves

Our goal here is to develop criteria that result in intersection free Catmull-Rom curves. There are three cases to consider: the *l*ocal case where we must avoid cusps and self-intersections within a single polynomial, the *a*djacent case where we consider intersection between adjacent polynomials, and the *g*lobal case where different polynomial segments not adjacent to one another may intersect.

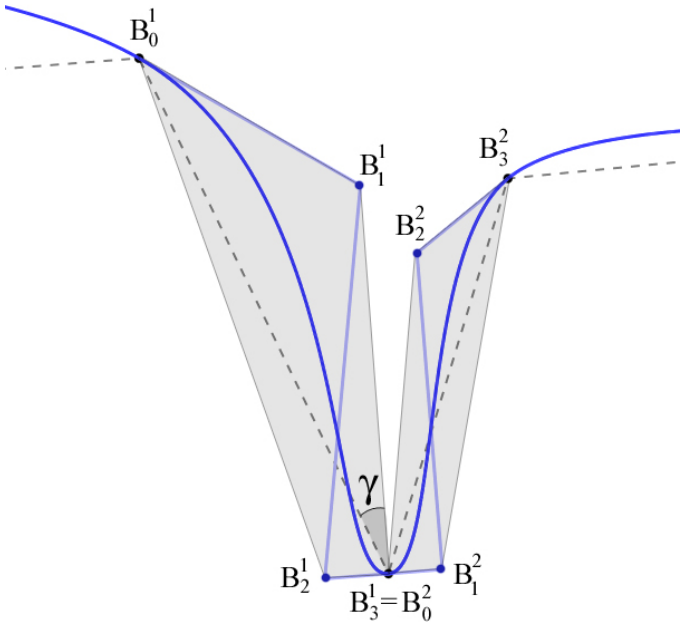


Figure 6: Bézier control polygons of two neighboring curve segments.

In Section 3 we showed that by using centripetal parameterization we can guarantee that the curve will not contain cusps or self-intersections within curve segments, satisfying the local case. Also, as shown in section 4, we have a bounding box that defines limits on the distance of the curve from the corresponding segment of the control polygon. As long as we use a centripetal parameterization and avoid overlapping bounding boxes, the curve will not self-intersect. This satisfies the global case.

Unfortunately, we cannot use the same bounding boxes to deal with the adjacent case, since bounding boxes of adjacent segments will always overlap. Therefore, we must have an alternative means of ensuring that such adjacent segments do not intersect. We do this by constructing an *angular bound* on the control polygon of the curve.

Consider the two Bézier curves in Figure 6 that have control points $\mathbf{B}_0^1, \mathbf{B}_1^1, \mathbf{B}_2^1, \mathbf{B}_3^1$ and $\mathbf{B}_0^2, \mathbf{B}_1^2, \mathbf{B}_2^2, \mathbf{B}_3^2$ where $\mathbf{B}_3^1 = \mathbf{B}_0^2$ corresponding to two different curve segments of the Catmull-Rom curve with centripetal parameterization. A Bézier curve lies within the convex hull of its control points, hence our intersection criteria will simply guarantee that the convex hulls do not intersect. Notice that each convex hull (shown shaded gray in the

figure) contains its end-points. Furthermore, we can exclude the points $\mathbf{B}_2^1$ and $\mathbf{B}_1^2$ as they are necessarily co-linear and will lie on opposite sides of the "V" formed by the control polygon.

Therefore, we need only consider the hull formed by $\mathbf{B}_0^1, \mathbf{B}_1^1, \mathbf{B}_3^1$ intersected with the hull formed by $\mathbf{B}_0^2, \mathbf{B}_2^2, \mathbf{B}_3^2$. There are three cases to consider: $\mathbf{B}_1^1$ and $\mathbf{B}_2^2$ both lie on the outside of the "V", $\mathbf{B}_1^1$ is on the inside of the "V" and $\mathbf{B}_2^2$ is on the outside (the symmetric case follows), and the case where both $\mathbf{B}_1^1$ and $\mathbf{B}_2^2$ are on the interior (as illustrated in Figure 6).

For the first case, the portion of the convex hull we need to avoid intersecting consists only of the edges of the control polygon. It is not possible to have a self-intersection in such cases, since the curves are bounded away from each other.

The other two cases are very similar, and so we will analyze the convex hull for only one side. We will bound the angle γ between $\mathbf{B}_1^1 - \mathbf{B}_3^1$ and $\mathbf{B}_0^1 - \mathbf{B}_3^1$. First, we compute the length of the projection of $\mathbf{B}_1^1 - \mathbf{B}_3^1$ onto $\mathbf{B}_0^1 - \mathbf{B}_3^1$. Using Equation 4, this length is given by

$$d_2 - d_2 \frac{r^{1-\alpha}\cos(\theta)}{3 + 3r^{\alpha}}.$$

The ratio involving this length and the distance of $\mathbf{B}_1^1$ to the line segment formed by $\mathbf{B}_0^1$ and $\mathbf{B}_3^1$ will be $\tan(\gamma)$. Combining this expression with Equation 5 when $\alpha = \frac{1}{2}$, we obtain

$$\tan(\gamma) = \frac{\sqrt{r}\,|\sin(\theta)|}{3 + 2\sqrt{r} - \sqrt{r}\cos(\theta)}.$$

Maximizing this function we find that $\gamma \leq \frac{\pi}{6}$; that is, the curve will extend beyond the control polygon toward the interior of the "V" within an angle of $\frac{\pi}{6}$.

Therefore, when both $\mathbf{B}_1^1$ and $\mathbf{B}_2^2$ are on the interior of the "V", the angle that will guarantee no intersection between adjacent curve segments is $\frac{\pi}{3}$. This bound, in combination with the global intersection test from Section 4, allows us to guarantee an intersection free Catmull-Rom curve when using centripetal parameterization.

To summarize, we form intersection-free curves as follows:

1. We use a centripetal parameterization to avoid self intersections within a curve segment.

2. To avoid intersections between adjacent curve segments, we restrict the angular bound of adjacent control polygon segments to be greater than $\frac{\pi}{3}$, as described in this section.
3. We avoid intersections between other curve segments by not allowing overlap between bounding boxes for non-adjacent segments.

6. Discussion

As a result of our theoretical and experimental analysis, we had several observations about Catmull-Rom curves. In this section we discuss some general intuition related to the use of various parameterizations on Catmull-Rom curves.

Note that all the parameterizations we consider are based on the distance between control points. Therefore, when all line segments of the control polygon have the same length, all parameterizations of this family produce the same curve. The differences between parameterization choices appear when the control polygon has line segments with different lengths. As the differences between the lengths of neighboring segments increase, the different characteristics of the parameterizations are amplified.

6.1. Distance to Control Polygon

In the previous sections we discussed the upper bound for the distance between the curve and the corresponding edge of the control polygon. In practice this distance can be much smaller than the upper bound. In fact, for uniform parameterization as an edge becomes larger compared to its neighbors, the curve becomes closer to the edge. A similar behavior happens with chordal parameterization for shorter edges, while longer edges push the curve segments of the chordal parameterization closer to the upper bound. In that sense, edge distance behavior of uniform and chordal parameterizations are the opposite of each other. This behavior can be seen in Figure 7.

With increasing α values, for shorter edges the curve rapidly deviates from uniform parameterization and approaches chordal parameterization curve slowly. On the other hand, for longer control polygon segments, as α increases, the curve slowly deviates from uniform parameterization and rapidly approaches chordal parameterization with large values of α. This behavior is demonstrated in Figure 7. Therefore, the result of centripetal parameterization is relatively closer to uniform parameterization for longer edges, and closer to chordal parameterization for shorter edges. As a result, curves with

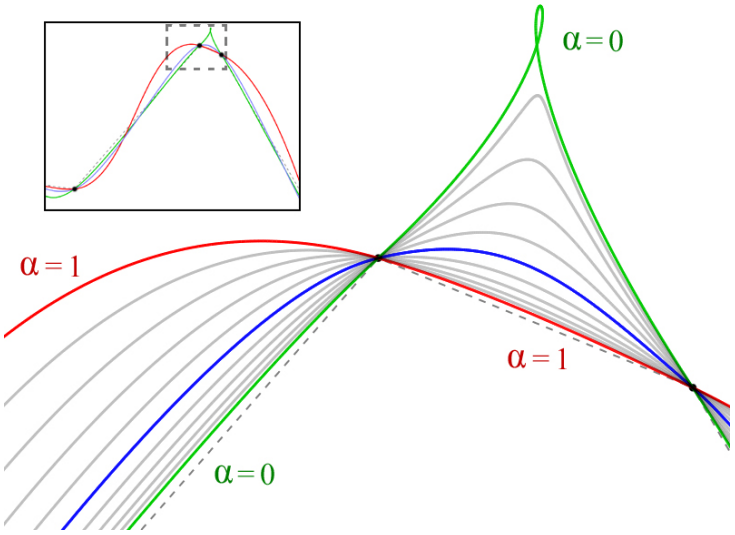


Figure 7: Cubic Catmull-Rom curves with parameterization values α ranging from 0 to 1. The green curve is $\alpha = 0$ (uniform), the blue curve is $\alpha = \frac{1}{2}$ (centripetal), the red curve is $\alpha = 1$ (chordal), and the grey curves are other values of α between 0 and 1 with regular intervals 0.1.

centripetal parameterization are closer to the control polygon than others when the entire curve is considered.

6.2. Cusps and Self-Intersections

Uniform parameterization often produces cusps or self-intersections within curve segments. Even when there are no cusps or intersections, uniform parameterization tends to produce high curvature points along shorter segments, which are usually undesirable in practice.

As α increases, such features become less likely to appear. As we show in Section 3, when $\alpha > \frac{1}{2}$ cusps or self-intersections can only happen when the Catmull-Rom curve overshoots its control points. However, centripetal parameterization is the only member of this parameterization family that guarantees no cusps or self-intersections anywhere within a single Catmull-Rom curve segment.

6.3. Edge Direction

The least favorable property of chordal parameterization is its extreme sensitivity to the direction of control polygon edges. This behavior can be

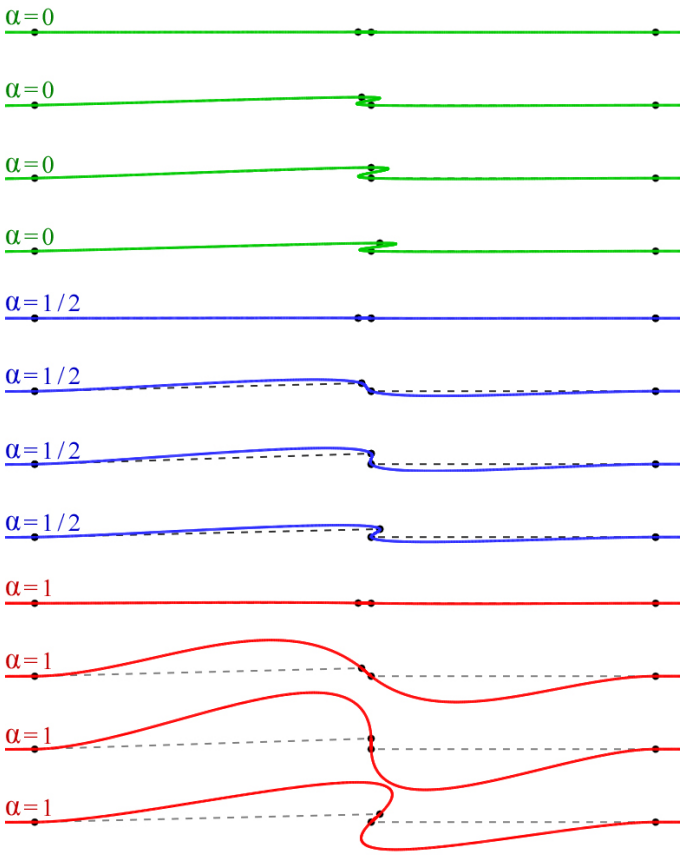


Figure 8: Cubic Catmull-Rom curves with parameterization values α ranging from 0 to 1. The green curve is $\alpha = 0$ (uniform), the blue curve is $\alpha = \frac{1}{2}$ (centripetal), the red curve is $\alpha = 1$ (chordal), and the grey curves are other values of α between 0 and 1 with regular intervals 0.1.

observed near short edges. While the curves with chordal parameterization are very close to shorter edges of the control polygon, this makes the curves overshoot when longer edges are adjacent to shorter ones. As a result, relatively minor changes in the position of a control point with a short edge can drastically alter the shape of the curve with chordal parameterization. This behavior is demonstrated in Figure 8. Note that uniform and centripetal parameterizations are not affected nearly as much.

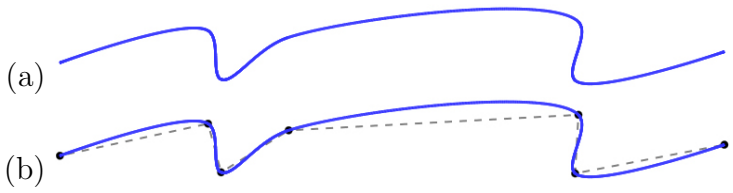


Figure 9: (a) A cubic Catmull-Rom curve with centripetal parameterization, (b) the same curve with its control polygon. Note that control points coincide with local high curvature points on the curve.

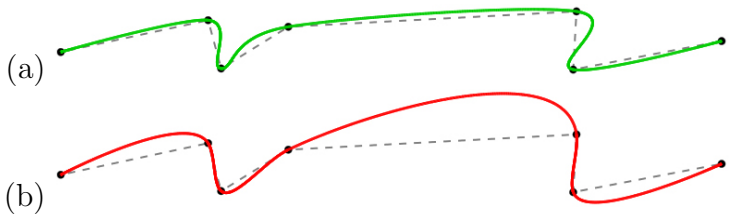


Figure 10: The same control points in Figure 9 with (a) uniform and (b) chordal parameterizations. Notice that local high curvature points do not coincide with the control points unlike centripetal parameterization in Figure 9.

6.4. Curvature

Cubic Catmull-Rom curves are not curvature continuous and have curvature discontinuities at the control points. However, the curvature is continuous within a single curve segment. In our experiments, we noticed that centripetal parameterization tends to produce the highest curvature within a curve segment at one of its end-points (Catmull-Rom control points). Unfortunately, this behavior is not guaranteed, and one can place control points in such a way as to demonstrate a counterexample. Despite this lack of a guarantee, counter-examples are difficult to find and, in most cases, the curvature does concentrate at the control points. We demonstrate this behavior in Figure 9. Note that the control points shown in Figure 9b correspond to local curvature maxima in Figure 9a. High curvature points generated with other parameterizations often do not coincide with control points (Figure 10). In practice, this lack of correspondence makes it significantly more difficult to control these curves to create a desired shape.

7. Applications

Catmull-Rom curves have a wide range of applications, particularly those involving interpolation of control points. The fact that they have local support and that they have a polynomial representation also make Catmull-Rom curves preferable over other curve formulations in many settings. In this section we discuss two application domains for Catmull-Rom curves and present how the choice of parameterization makes a significant difference in those applications.

7.1. Animation Curves

There are several applications, ranging from robotics to computer animation, in which a set of parameters x need to be interpolated over time. The parameters may be joint angles, positions, or even higher order terms such as velocity. These parameters are specified at particular points in time, and it is crucial to have a smooth interpolation of these values. We will refer to these interpolations, generally, as animation curves. Borrowing the language of computer animation, we will refer to the parameters as animation parameters, and a specific specification of parameters to be interpolated at a particular time as a key-frame. Catmull-Rom curves are already popular for creating animation curves, but they may produce highly undesirable results if the parameterization is not chosen properly.

An animation curve is a function $x = F(t)$, which produces a set of animation parameters x at the given time t. We define a Catmull-Rom spline $C(s)$ to represent the curve $F(t)$ where our curve has control points $P_i = (t_i, x_i)$ defined by the value x_i at time t_i for the i^{th} key-frame. However, when it comes to the parameterization of the Catmull-Rom curve, we cannot simply use the Euclidean distance between control points P_i, because x and t in general have different units. Since the Euclidean distance that combines two unrelated units is not a meaningful measure, we use the distances between key-frame times t_i and ignore the values x_i for parameterization.

In this notation the Catmull-Rom curve $C(s)$ can be written as two curves, such that $x = X(s)$ and $t = T(s)$. Since we only use t_i values for building the parameterization s_i for each key-frame point, in effect the parameterization is computed for $T(s)$ only and the same parameterization is used for $X(s)$. Using this notation,

$$F(t) = X\left(T^{-1}(t)\right). \tag{9}$$

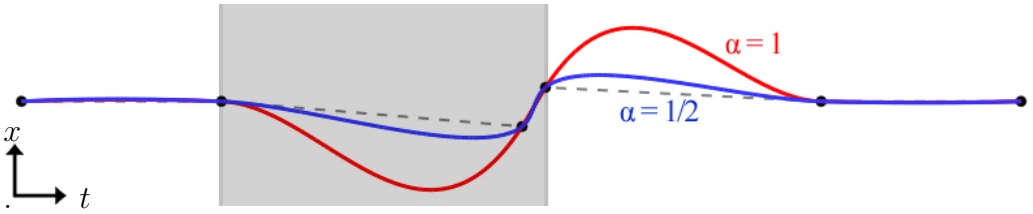


Figure 11: Chordal ($\alpha = 1$) and centripetal ($\alpha = \frac{1}{2}$) Catmull-Rom curves that interpolate the same key-frames. The two close-by key-frames in the middle cause the chordal curve to overshoot and highly deviate from the control polygon, while the centripetal curve deviates significantly less. The shaded segments of these curves are used for driving the animation in Figure 12.

Notice that for $T(s)$ to have an inverse, $t = T(s)$ must be one-to-one and onto over the time range of the animation. In effect (since this is a curve in one dimension), we want $T(s)$ to be monotonic. Notice that if $T(s)$ is not monotonic, then the results would be meaningless in terms of animation, since there would essentially be two or more values of x for a given value of t. We ensure that $T(s)$ meets this criterion by ensuring that the Catmull-Rom curve we use for representing $T(s)$ does not have self-intersections. We know that for general Catmull-Rom curves, only centripetal parameterization guarantees no self-intersections within a curve segment. However, animation curves are a special case, because each t_i has to be strictly increasing (i.e. $t_i < t_{i+1}$). We therefore examine this case in more detail.

For monotonically increasing t_i, $\cos(\theta) = -1$ in Equation 3 and so Equation 3 can never be satisfied regardless of the value of α. Hence, the curve $T(s)$ cannot have a negative derivative at its end-points (if the derivative were negative, the curve could not be monotonically increasing). However, $T(s)$ may have an interior cusp if the curve does not meet Theorem 2. Unfortunately, for $\alpha \in [0, \frac{1}{2})$, Equation 4 may be satisfied and an interior cusp may exist, meaning that $T(s)$ is not invertible. This fact implies that building such animation curves with $\alpha < \frac{1}{2}$ is not possible in general.

Nevertheless, for $\alpha \in [\frac{1}{2}, 1]$, monotonic t_i imply that a monotonic curve $T(s)$ and $T^{-1}(t)$ will always exist. Moreover, since $T(s)$ is monotonic and interpolates the t_i, finding $T^{-1}(t)$ can be performed via a simple bisection over the parameter interval $t_i \leq t \leq t_{i+1}$. Finally, for the special case of chordal parameterization ($\alpha = 1$), $t = T(s) = s$ since Catmull-Rom curves have linear precision and the inverse simplifies in Equation 9.

We are then left with the question of whether any particular parameter-

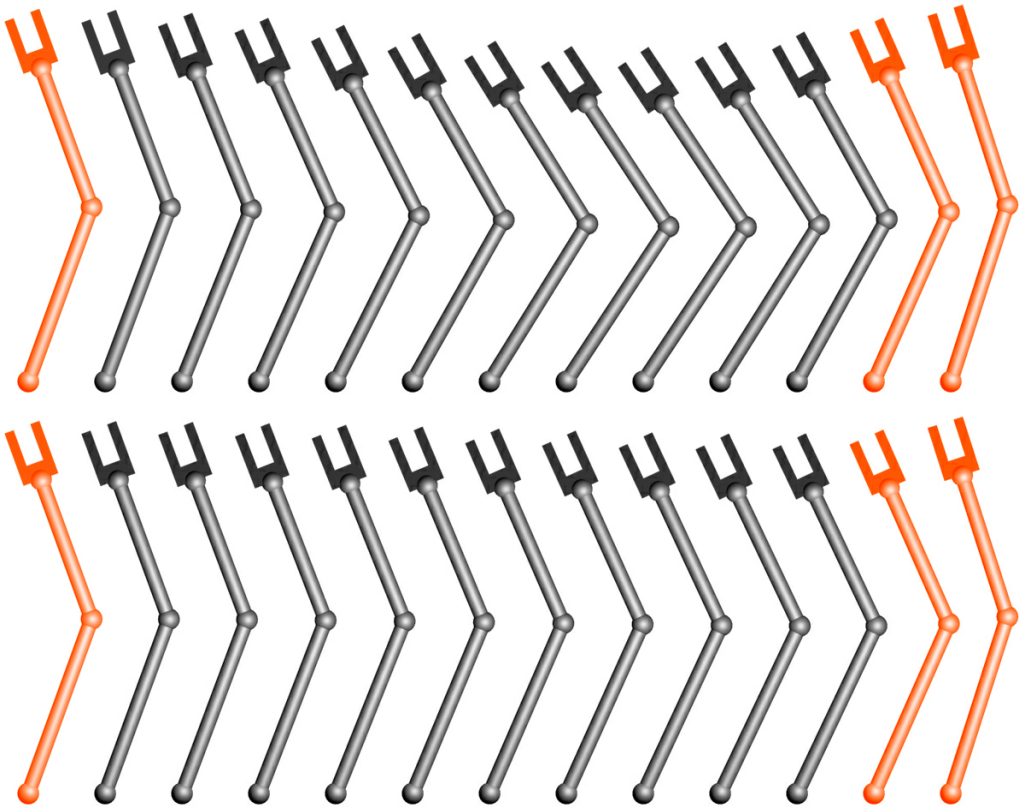


Figure 12: Animation of a robot arm comparing the interpolation generated by chordal ($\alpha = 1$) and centripetal ($\alpha = \frac{1}{2}$) Catmull-Rom curves. The first and the last two frames are key-frames and the rest are interpolated. This animation corresponds to the shaded segment of the animation curves in Figure 11. Notice that the interpolation with chordal parameterization highly deviates from user defined key-frame poses.

izations in the range $\alpha \in [\frac{1}{2}, 1]$ are better than others. In our experiments with animation curves using various values of α we observed that as α gets larger, the resulting Catmull-Rom curve deviates further away from the control polygon of the curve (the straight lines that connect consecutive control points) for segments that interpolate distant key-frames. This is equivalent to saying that the interpolated animation will deviate more from the user-defined key-frame values. This observation is consistent with our discussions in the previous sections. As the α value gets closer to 1, the resulting animation curve becomes more sensitive to the direction of control polygon edges that are shorter than their neighbors. That is, closely-placed key frames

will tend to have a greater impact (creating more deviation) on interpolated frames farther from the key-frames.

Figure 11 demonstrates this effect by showing animation curves that interpolate a set of key-frame values using Catmull-Rom curves with $\alpha = \frac{1}{2}$ and $\alpha = 1$. As can be seen from this figure, chordal parameterization ($\alpha = 1$) produces curves that deviate further when interpolating distant key-frames. We also used a segment of these curves to derive an animation of a robot arm, shown in Figure 12. The interpolated frames in Figure 12 show how much more exaggerated the motion is with chordal parameterization as compared to centripetal parameterization.

7.2. Path Curves

Another possible application for Catmull-Rom splines are path curves that define the motion path of an object in 3D space. These curves can arise in multiple domains where the position/configuration of a device is specified at certain points. For example, in robotics, probabilistic roadmaps [11] represent a path as connected points along a graph, but the paths are typically not smooth. Catmull-Rom splines provide a simple method for creating smooth curves that follow the discovered path. Likewise, tool path generation may require a smooth path that interpolates various 3D points on a machined surface; Catmull-Rom splines can provide such a curve.

Just like animation curves, path curves are also defined by a number of key-frame positions and the curve must interpolate these key-frame points. Unlike animation curves, however, path curves do not specify time values and geometrically simply represent a curve in the space of animation parameters x.

Loops and self-intersecting curve segments are often undesirable path curves. Moreover, key-frames are generally used for representing the extreme positions of a motion. Therefore, it is preferable that the curve does not overshoot the key-frames. As we have shown, only centripetal parameterization of Catmull-Rom curves guarantees these conditions.

When centripetal parameterization is used with Catmull-Rom splines to define a path curve, the direction of motion for the object following this path will always be towards the next key-frame position. Let P_1 and P_2 be the two consecutive key-frame positions that a curve segment interpolates. The segment of the Catmull-Rom curve between P_1 and P_2 is guaranteed to be in the same direction as the line connecting P_1 and P_2 (the dot product of the derivative of the curve and $P_2 - P_1$ will be positive) when using centripetal

parameterization. Therefore, the object always moves from one control point towards the next one and never in the opposite direction.

Note that when using Catmull-Rom splines for representing path curves, these curves only represent the shape of the path and do not provide a positions for a given time t like animation curves do. Instead, these curves focus more on the shape of the curve. To provide a desired speed along the curve, we typically must reparameterize the curve. The natural parameterization of the curve provided by s will produce a wide range of speeds along the curve for different values of the parameterization constant α. The desired reparameterization of the curve may not have an analytical solution (for example arc-length parameterization); however, most reparameterizations can be easily computed numerically.

8. Conclusion

Our analysis on the parameterization of cubic Catmull-Rom curves demonstrates that centripetal parameterization has special properties. In particular, this parameterization was the only parameterization that guaranteed no local self-intersections of the curve. Furthermore, we created distance bounds of the curve to its control polygon for curves within this parameterization family. Using these distance bounds, we derived angle constraints on the control polygon that could guarantee Catmull-Rom curves with centripetal parameterization were globally intersection free. Finally, these properties were valid in general dimension $\mathbb{R}^m$.

Currently, we have only explored C^1 Catmull-Rom curves. While these curves are by far the most popular in the family of Catmull-Rom curves, we have yet to examine higher degree curves. One important property that is lost with higher degree Catmull-Rom curves is the lack of local self-intersections with centripetal parameterization as we can create cases where this phenomenon happens even using C^2 Catmull-Rom curves. However, the frequency of such self-intersection seems to be less than with other parameterizations, but it is unclear whether anything precise can be said about this property when using higher order continuity.

9. Acknowledgments

This work was funded in part by NSF grant CCF-07024099.

References

[1] N. Dyn, M. S. Floater, K. Hormann, Four-point curve subdivision based on iterated chordal and centripetal parameterizations, Computer Aided Geometric Design 26 (3) (2009) 279–286.

[2] M. P. Epstein, On the influence of parametrization in parametric interpolation, SIAM Journal on Numerical Analysis 13 (2) (1976) 261–268.

[3] M. S. Floater, T. Surazhsky, Parameterization for curve interpolation, in: Topics in multivariate approximation and interpolation, 2006, pp. 39–54.

[4] E. T. Y. Lee, Choosing nodes in parametric curve interpolation, Computer Aided Design 21 (6) (1989) 363–370.

[5] M. S. Floater, On the deviation of a parametric cubic spline interpolant from its data polygon, Computer Aided Geometric Design 25 (3) (2008) 148–156.

[6] E. Catmull, R. Rom, A class of local interpolating splines, Computer Aided Geometric Design (1974) 317–326.

[7] P. J. Barry, R. N. Goldman, A recursive evaluation algorithm for a class of catmull-rom splines, SIGGRAPH Computer Graphics 22 (4) (1988) 199–204.

[8] T. A. Foley, G. M. Nielson, Knot selection for parametric spline interpolation (1989) 261–272.

[9] G. Nielson, T. Foley, A survey of applications of an affine invariant metric (1989) 445–468.

[10] D. Manocha, J. F. Canny, Detecting cusps and inflection points in curves, Computer Aided Geometric Design 9 (1) (1992) 1–24.

[11] L. E. Kavraki, P. Svestka, J.-C. Latombe, M. Overmars, Probabilistic roadmaps for path planning in high dimensional configuration spaces, IEEE Transactions on Robotics and Automation 12 (4) (1996) 566–580.