Connected Minimal Acceleration Trigonometric Curves

Tony Barrera [*]
Barrera Kristiansen AB

Anders Hast [†]
Creative Media Lab,
University of Gävle

Ewert Bengtsson [‡]
Centre for Image Analysis
Uppsala University

Abstract

We present a technique that can be used to obtain a series of connected minimal bending trigonometric splines that will intersect any number of predefined points in space. The minimal bending property is obtained by a least square minimization of the acceleration. Each curve segment between two consecutive points will be a trigonometric Hermite spline obtained from a Fourier series and its four first terms. The proposed method can be used for a number of points and predefined tangents. The tangent length will then be optimized to yield a minimal bending curve. We also show how both the tangent direction and length can be optimized to give as smooth curves as possible. It is also possible to obtain a closed loop of minimal bending curves. These types of curves can be useful tools for 3D modelling, etc.

Keywords: Trigonometric curves, Hermite curves, least square minimization

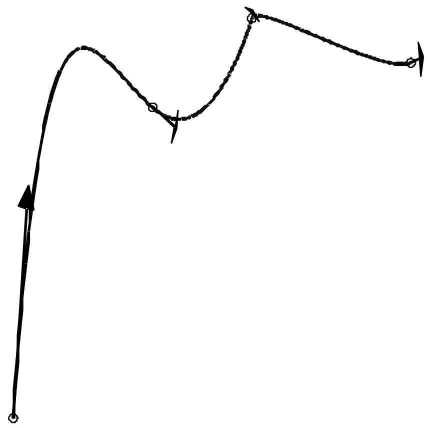

Figure 1: Multiple connected trigonometric curve with non optimal predefined tangents.

1 Introduction

This paper proposes a simple technique that will make it possible to construct a minimal bending curve through a number of consecutive points in space, using trigonometric splines [Schoenberg 1964]. Thus, each curve consists of a number of connected trigonometric Hermite spline segments [Alba-Fernandez 2004]. Each spline will start in one predefined point and end in the consecutive point, and the next curve segment will start in that point and end in the next point, and so forth.

In [Barrera 2005] a similar technique is presented where a minimal bending cubic curve is obtained where both the points and the directions at these points are given. That algorithm will compute optimal tangent lengths. Bartels et al [Bartels 1998] show how a minimal bending cubic curve can be obtained using the points only as constraints for the curve. The resulting splines will be Hermite splines and should not be confused with Catmul-Rom splines [Catmull 1974] which also intersect the given points. However, they are constructed in a quite different way.

Figure 1 shows a trigonometric Hermite curve where four points and tangents are set as constraints. The left part of the curve has rather large tangents, which make the curve bend heavily around the intersection points. On the right side the tangents are rather short, which makes the curve bend rapidly around the intersection points. A minimal bending curve will have minimal acceleration over the curve and this will make the curve smoother. Note that the length of the tangents have been scaled down to 25% in all the figures so that the tangents will not be too large compared to the curve.

Several such curves will be presented in this paper. First we will prove that a cubic curve that only has points as constraints will have this minimal acceleration property. The derivation will serve as an example when we proceed to discuss trigonometric curves instead. These curves have the advantage that they can define everything from straight lines to perfect circles. Next we will show how a trigonometric curve using points as constraints can be obtained and then we will show how a curve using both points and tangent directions can be constructed. In the latter case the tangent is set to an optimal, while in the first case both tangent length and direction is set to an optimal. Hence this type of curve will always be smoother, but we loose the possibility to determine direction in each point, which might be desirable for camera movements [Vlachos 2001] etc.

[*]e-mail: tony.barrera@spray.se
[†]e-mail: aht@hig.se
[‡]e-mail: ewert@cb.uu.se

2 Least Square Minimization of Cubic Hermite Curves

We will start by proving that a cubic Hermite curve [Hearn 2004] that intersects a number of given points will actually have the minimal acceleration property. This will serve as an example of how the minimal acceleration is obtained since the equations are shorter and easier to understand than for trigonometric curves. Then we will go on to give examples of how this works for trigonometric curves.

The total curvature of a curve $f(t)$ in the parametric interval $[0,1]$ of one single curve segment is defined by

$$\int_0^1 \|\kappa(t)\| dt \tag{1}$$

Where $\kappa(t)$ is the curvature of the curve at t. This formula often causes very complex expressions, so it is more common to use the integral

$$\int_0^1 \|\mathbf{f}''(t)\|^2 dt \tag{2}$$

This integral sums the acceleration, i.e. the square of the second derivative over the curve. The acceleration is minimized by differentiating on some variable and set the result to zero so that the minimum is obtained. This is the essence of least square minimization [Burden 1989]. In our case we would like to find the optimal tangents that will give a minimal bending curve. If there are $k+1$ number of points, then there will be k number of curve segments. Hence we differentiate on the tangents and solve

$$\frac{\partial}{\partial \mathbf{T}_i} \int_0^1 \|\mathbf{f}_1''(t)\|^2 + \|\mathbf{f}_2''(t)\|^2 + ... + \|\mathbf{f}_{k+1}''(t)\|^2 dt = 0 \tag{3}$$

where $i = 1, 2, ..., k+1$.

In order to be able to derive the curve we must first compute the second derivatives of the Hermite curve. A general cubic curve is defined by

$$\mathbf{f}(t) = \mathbf{A}t^3 + \mathbf{B}t^2 + \mathbf{C}t + \mathbf{D} \tag{4}$$

and a Hermite curve has the initial conditions

$$\mathbf{f}(0) = \mathbf{P}_i \tag{5}$$
$$\mathbf{f}(1) = \mathbf{P}_{i+1} \tag{6}$$
$$\mathbf{f}'(0) = \mathbf{T}_i \tag{7}$$
$$\mathbf{f}'(1) = \mathbf{T}_{i+1} \tag{8}$$

Where $\mathbf{T}_i$ and $\mathbf{T}_{i+1}$ are two tangent vectors to be determined for minimum acceleration. The Hermite [Hearn 2004] curve is defined by solving the system

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \\ \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_i \\ \mathbf{P}_{i+1} \\ \mathbf{T}_i \\ \mathbf{T}_{i+1} \end{bmatrix} \tag{9}$$

The solution of this system is

$$\mathbf{A} = \mathbf{T}_i + \mathbf{T}_{i+1} - 2\mathbf{P}_{i,i+1} \tag{10}$$
$$\mathbf{B} = 3\mathbf{P}_{i,i+1} - 2\mathbf{T}_i - \mathbf{T}_{i+1} \tag{11}$$
$$\mathbf{C} = \mathbf{T}_i \tag{12}$$
$$\mathbf{D} = \mathbf{P}_i \tag{13}$$

where $\mathbf{P}_{i,i+1} = \mathbf{P}_{i+1} - \mathbf{P}_i$.

Hence

$$\mathbf{f}''(t) = 6\mathbf{A}t + 2\mathbf{B} \tag{14}$$

and

$$\|\mathbf{f}''(t)\|^2 = 36\mathbf{A}^2 t^2 + 24\mathbf{A}\cdot\mathbf{B}t + 4\mathbf{B}^2 \tag{15}$$

where we use the notation $\mathbf{A}^2 = \mathbf{A}\cdot\mathbf{A}$ and so forth. Substituting equations (10) through (13) into equation (15) and differentiating on $\mathbf{T}_1$ as in equation (3) gives

$$\frac{\partial}{\partial \mathbf{T}_1} \int_0^1 \|\mathbf{f}''(t)\|^2 dt = 8\mathbf{T_1} + 4\mathbf{T_2} - 12\mathbf{P_{12}} \tag{16}$$

Moreover we have

$$\frac{\partial}{\partial \mathbf{T}_2} \int_0^1 \|\mathbf{f}''(t)\|^2 dt = 4\mathbf{T_1} + 16\mathbf{T_2} + 4\mathbf{T_3} - 12\mathbf{P_{12}} - 12\mathbf{P_{23}} \tag{17}$$
$$\frac{\partial}{\partial \mathbf{T}_3} \int_0^1 \|\mathbf{f}''(t)\|^2 dt = 4\mathbf{T_2} + 16\mathbf{T_3} + 4\mathbf{T_4} - 12\mathbf{P_{23}} - 12\mathbf{P_{34}} \tag{18}$$

and finally

$$\frac{\partial}{\partial \mathbf{T}_{k+1}} \int_0^1 \|\mathbf{f}''(t)\|^2 dt = 8\mathbf{T_k} + 4\mathbf{T_{k+1}} - 12\mathbf{P_{k,k+1}} \tag{19}$$

Next we set each equation equal to zero and solve for each tangent. After dividing each equation by four this yields a system of equations

$$\begin{bmatrix} 2 & 1 & 0 & 0 & ... & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & ... & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & ... & 0 & 0 & 0 \\ . & . & . & . & ... & . & . & . \\ . & . & . & . & ... & . & . & . \\ . & . & . & . & ... & . & . & . \\ 0 & 0 & 0 & 0 & ... & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \\ \mathbf{T}_3 \\ . \\ . \\ . \\ \mathbf{T}_{k+1} \end{bmatrix} = \begin{bmatrix} 3\mathbf{P}_{12} \\ 3(\mathbf{P}_{12} + \mathbf{P}_{23}) \\ 3(\mathbf{P}_{23} + \mathbf{P}_{34}) \\ . \\ . \\ . \\ 3\mathbf{P}_{k,k+1} \end{bmatrix} \tag{20}$$

A system involving a matrix of this form is called a *tridiagonal system* and can be solved efficiently using a specialized algorithm [Lengyel 2004]. This is the same system, which is derived in [Bartels 1998]. However, they derive it in a different way were they set up a system requiring C^2 continuity at the intersection points. Nevertheless, our derivation proves that this type of curve have the minimal acceleration property.

3 Trigonometric Hermite splines

Trigonometric splines (or trigonometric polynomials) were introduced by Schoenberg [Schoenberg 1964] and have been investigated extensively in math and computer aided geometry literature, [Walz 1997], [Lyche 1979], [Han 2003], just to mention a few. However, they have not gained much interest in computer graphics. One reason is probably that it involves the computation of trigonometric functions and those have been computationally expensive. With faster hardware they may gain the interest from the computer graphics community as a modelling tool, since it is possible to construct everything from straight lines to perfect circle arcs. The latter is impossible with cubic curves.

A trigonometric spline can be constructed from a truncated Fourier series [Schoenberg 1964], [Walz 1997]. An Hermite spline is defined by two points and the tangents in these points and therefore we have four constraints and thus we need four terms in the Fourier series. The trigonometric curve is therefore defined as

$$\mathbf{f}(\theta) = \mathbf{a} + \mathbf{b}\cos\theta + \mathbf{c}\sin\theta + \mathbf{d}\cos 2\theta \tag{21}$$

Using the conditions in (5) through (8), the curve is found by solving

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{T}_1 \\ \mathbf{T}_2 \end{bmatrix} \tag{22}$$

The solution is

$$\mathbf{a} = \frac{1}{2}(\mathbf{P}_1 + \mathbf{P}_2 - \mathbf{T}_1 + \mathbf{T}_2) \tag{23}$$

$$\mathbf{b} = -\mathbf{T}_2 \tag{24}$$

$$\mathbf{c} = \mathbf{T}_1 \tag{25}$$

$$\mathbf{d} = \frac{1}{2}(\mathbf{P}_1 - \mathbf{P}_2 + \mathbf{T}_1 + \mathbf{T}_2) \tag{26}$$

By forcing $\mathbf{d}$ to be equal to zero in equation (21) we get

$$\mathbf{f}(\theta) = \mathbf{a} + \mathbf{b}\cos\theta + \mathbf{c}\sin\theta \tag{27}$$

This is obviously the equation for a circle and this proves that it is possible to construct a perfect circle arc using these curves. Since the curve is parametric it is easy to see that it is possible to construct straight lines using the trigonometric splines. The coefficients are vectors and the function produces a point in space and each coordinate has its own expression and the only thing that differs is the coefficients, and therefore it is no problem to construct a straight line even though trigonometric functions are involved.

4 Least Square Minimization of Trigonometric Hermite Splines

Once again we use equation (3) in order to optimize both tangent length and direction for the Trigonometric Hermite spline defined in equation (21). This will yield a system of equations that must be solved.

$$\begin{bmatrix} A & 2B & 0 & 0 & \dots & 0 & 0 & 0 \\ B & A & B & 0 & \dots & 0 & 0 & 0 \\ 0 & B & A & B & \dots & 0 & 0 & 0 \\ . & . & . & . & \dots & . & . & . \\ . & . & . & . & \dots & . & . & . \\ . & . & . & . & \dots & . & . & . \\ 0 & 0 & 0 & 0 & \dots & 0 & 2B & A \end{bmatrix} \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \\ \mathbf{T}_3 \\ . \\ . \\ . \\ \mathbf{T}_{k+1} \end{bmatrix} = \begin{bmatrix} 2C\mathbf{P}_{12} \\ C(\mathbf{P}_{12}+\mathbf{P}_{23}) \\ C(\mathbf{P}_{23}+\mathbf{P}_{34}) \\ . \\ . \\ . \\ 2C\mathbf{P}_{k,k+1} \end{bmatrix} \tag{28}$$

where we have

$$A = 15\pi - 16 \tag{29}$$

$$B = 6\pi - 11 \tag{30}$$

$$C = 6\pi - 4 \tag{31}$$

Figure 2 shows how the proposed approach will yield a curve that is much smoother than the curve in figure 1, since both the tangent direction and length are set to an optimal, giving a minimal bending curve.

4.1 Optimal tangent length

If we want our curves to have the same direction as the tangents in the intersection points, then we can change the computation so that we solve for optimal tangent length only instead of solving for both optimal tangent length and direction. In this case we introduce α_i as the length of each tangent $\mathbf{T_i}$.

The equation now becomes

$$\frac{\partial}{\partial \alpha_i} \int_0^1 \|\mathbf{f}_1''(t)\|^2 + \|\mathbf{f}_2''(t)\|^2 + \dots + \|\mathbf{f}_{k+1}''(t)\|^2 dt = 0 \tag{32}$$

where $i = 1, 2, \dots, k+1$.

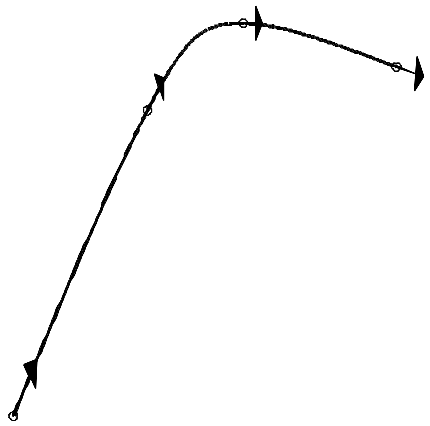

Figure 2: Multiple connected minimal acceleration trigonometric curves, with both optimal tangent direction and length.

The resulting coefficient matrix is

$$\begin{bmatrix} A\mathbf{T}_1^2 & 2B\mathbf{T}_1\cdot\mathbf{T}_2 & 0 & 0 & \dots & 0 & 0 & 0 \\ B\mathbf{T}_1\cdot\mathbf{T}_3 & A\mathbf{T}_2^2 & B\mathbf{T}_2\cdot\mathbf{T}_3 & 0 & \dots & 0 & 0 & 0 \\ 0 & \mathbf{T}_2\cdot\mathbf{T}_3 & A\mathbf{T}_3^2 & B\mathbf{T}_3\cdot\mathbf{T}_4 & \dots & 0 & 0 & 0 \\ . & . & . & . & \dots & . & . & . \\ . & . & . & . & \dots & . & . & . \\ . & . & . & . & \dots & . & . & . \\ 0 & 0 & 0 & 0 & \dots & 0 & 2B\mathbf{T}_k\cdot\mathbf{T}_{k+1} & A\mathbf{T}_{k+1}^2 \end{bmatrix} \tag{33}$$

And the variables to solve for are

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ . \\ . \\ . \\ \alpha_{k+1} \end{bmatrix} \tag{34}$$

Finally the column of constants is

$$\begin{bmatrix} 2C\mathbf{T}_1\cdot\mathbf{P}_{12} \\ C\mathbf{T}_2\cdot(\mathbf{P}_{12}+\mathbf{P}_{23}) \\ C\mathbf{T}_3\cdot(\mathbf{P}_{23}+\mathbf{P}_{34}) \\ . \\ . \\ . \\ 2C\mathbf{t}_{k+1}\cdot\mathbf{P}_{k,k+1} \end{bmatrix} \tag{35}$$

In figure 3 it is clear that the tangents have the same directions as in figure 1. However, the tangents have optimal length and the curve is thus smoother.

5 A Closed Loop

It is possible to connect any number of minimal acceleration trigonometric curves together into a closed loop as shown in figure 4. The end point for the last segment is set to be the same as the start point for the first segment. Likewise, the tangents at this

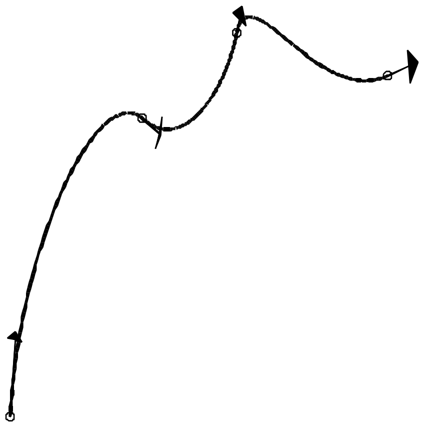

Figure 3: Multiple connected minimal acceleration trigonometric curve.

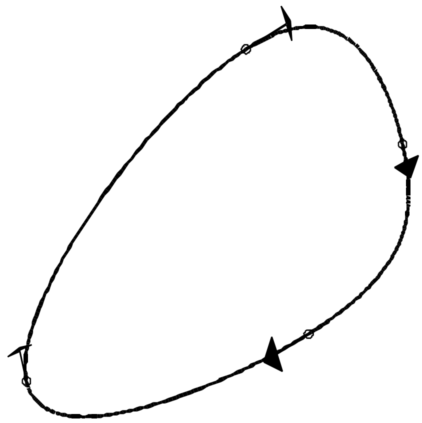

Figure 4: A closed loop of a trigonometric curve with optimal tangent length and direction

point is set in the same way. The equation to solve is now changed so that we have

$$\frac{\partial}{\partial \mathbf{T}_i} \int_0^1 \|\mathbf{f}_1''(t)\|^2 + \|\mathbf{f}_2''(t)\|^2 + ... + \|\mathbf{f}_k''(t)\|^2 dt = 0 \tag{36}$$

where $i = 1, 2, ..., k$. Note that this time there are k number of points and k number of curve segments.

This yields the following system

$$\begin{bmatrix} A & B & 0 & 0 & ... & 0 & 0 & B \\ B & A & B & 0 & ... & 0 & 0 & 0 \\ 0 & B & A & B & ... & 0 & 0 & 0 \\ . & . & . & . & ... & . & . & . \\ . & . & . & . & ... & . & . & . \\ . & . & . & . & ... & . & . & . \\ B & 0 & 0 & 0 & ... & 0 & B & A \end{bmatrix} \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \\ \mathbf{T}_3 \\ . \\ . \\ . \\ \mathbf{T}_k \end{bmatrix} = \begin{bmatrix} C(\mathbf{P}_{k,1} + \mathbf{P}_{12}) \\ C(\mathbf{P}_{12} + \mathbf{P}_{23}) \\ C(\mathbf{P}_{23} + \mathbf{P}_{34}) \\ . \\ . \\ . \\ C(\mathbf{P}_{k-1,k} + \mathbf{P}_{k,1}) \end{bmatrix} \tag{37}$$

where we have

$$A = 15\pi - 16 \tag{38}$$

$$B = 6\pi - 11 \tag{39}$$

$$C = 6\pi - 4 \tag{40}$$

The presence of the nonzero entries in the lower-left and upper-right corners make this system a *cyclic tridiagonal system*. It can also be solved efficiently [Press 1992].

5.1 Optimal Tangent Length

Now we proceed to show how a closed loop can be constructed when we want a specific tangent direction in each point. The equation to solve is

$$\frac{\partial}{\partial \alpha_i} \int_0^1 \|\mathbf{f}_1''(t)\|^2 + \|\mathbf{f}_2''(t)\|^2 + ... + \|\mathbf{f}_k''(t)\|^2 dt = 0 \tag{41}$$

where $i = 1, 2, ..., k$.

The coefficient matrix now becomes

$$\begin{bmatrix} A\mathbf{T}_1^2 & B\mathbf{T}_1 \cdot \mathbf{T}_2 & 0 & 0 & ... & 0 & 0 & B\mathbf{T}_1 \cdot \mathbf{t}_k \\ B\mathbf{T}_1 \cdot \mathbf{T}_3 & A\mathbf{T}_2^2 & B\mathbf{T}_2 \cdot \mathbf{T}_3 & 0 & ... & 0 & 0 & 0 \\ 0 & \mathbf{T}_2 \cdot \mathbf{T}_3 & A\mathbf{T}_3^2 & B\mathbf{T}_3 \cdot \mathbf{T}_4 & ... & 0 & 0 & 0 \\ . & . & . & . & ... & . & . & . \\ . & . & . & . & ... & . & . & . \\ . & . & . & . & ... & . & . & . \\ B\mathbf{T}_1 \cdot \mathbf{T}_k & 0 & 0 & 0 & ... & 0 & 2B\mathbf{T}_{k-1} \cdot \mathbf{T}_k & A\mathbf{T}_k^2 \end{bmatrix} \tag{42}$$

And the variables to solve for are

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ . \\ . \\ . \\ \alpha_k \end{bmatrix} \tag{43}$$

Finally the column of constants is

$$\begin{bmatrix} C\mathbf{T}_1 \cdot (\mathbf{P}_{12} + \mathbf{P}_{k,1}) \\ C\mathbf{T}_2 \cdot (\mathbf{P}_{12} + \mathbf{P}_{23}) \\ C\mathbf{T}_3 \cdot (\mathbf{P}_{23} + \mathbf{P}_{34}) \\ . \\ . \\ . \\ C\mathbf{T}_k \cdot (\mathbf{P}_{k-1,k} + \mathbf{P}_{k,1}) \end{bmatrix} \tag{44}$$

In figure 5 the tangent directions have been predefined. The curve is made smooth by the proposed minimal acceleration technique, so that the tangent length is set to an optimal.

6 Conclusions

We have presented a method that can be used to obtain minimal bending trigonometric Hermite curves, which can have a number of different constraints, like intersection points and tangent directions. These curves can be used in a number of areas, such as 3D-modeling and camera movements.

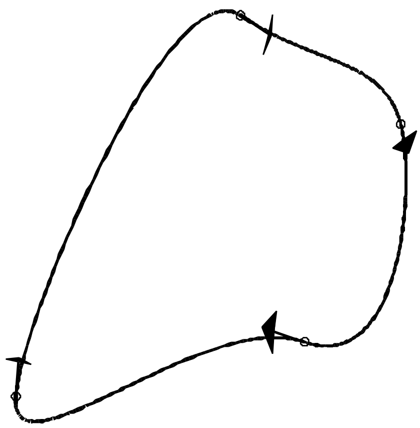

Figure 5: A closed loop with optimal tangent length for predefined tangent directions.

References

V. ALBA-FERNANDEZ, M.J. IBANEZ-PEREZ, M.D. JIMENEZ-GAMERO 2004. *A Bootstrap Algorithm for the Two-Sample Problem Using Trigonometric Hermite Spline Interpolation* Communications in Nonlinear Science and Numerical Simulation. Vol. 9. Num. 2. Pag. 275-286

T. BARRERA, A. HAST, E. BENGTSSON 2005. *Minimal Acceleration Hermite Curves* Graphics Programming Gems V, Charles River Media, Edited by Kim Pallister, pp 225-231.

R.H. BARTELS, J.C. BEATTY,AND B.A. BARSKY 1998. *An Introduction to. Splines for use in Computer Graphics and. Geometric Modeling* "Hermite and Cubic Spline Interpolation." Ch. 3, pp. 9-17.

R. L. BURDEN, J.D. FAIRES 1989. *Numerical Analysis* Numerical Analysis, PWS-KENT Publishing company Boston, pp. 439, 440.

E. CATMULL, R. ROM 1974. *A Class of Local Interpolating Splines* Computer Aided Geometric design, pp. 317-326.

A. FOLEY, J. D., ET AL 1997. *Computer Graphics: Principles and Practice, 2nd ed.* Addison-Wesley, p. 480.

X. HAN 2003. *Piecewise Quadratic Trigonometric Polynomial Curves* Mathematics of Computation, pp. 1369-1377.

D. HEARN, M.P BAKER 2004. *Computer Graphics with OpenGL* Pearson Education Inc., pp. 426-429.

E. LENGYEL. 2004. *E. Lengyel, Mathematics for 3D Game Programming and Computer Graphics, 2nd ed.* E. Lengyel, Charles River Media, pp. 433-436.

T. LYCHE. 1979. *A Newton form for Trigonometric Hermite Interpolation* E. Lengyel, Charles River Media, pp. 433-436.

PRESS ET AL. 1992. *Numerical Recipes in C.* [Press92] Cambridge University Press, pp. 74-75.

I. J. SCHOENBERG 1964. *On Trigonometric Spline Interpolation* Journal of Mathematics and Mechanics, pp. 795-825.

A. VLACHOS, J ISIDORO *Smooth C^2 Quaternion-based Flythrough Paths* Game Programming Gems 2, Charles River Media, Edited by Mark Deloura, pp. 220-227.

G. WALZ *Identities for Trigonometric B-Splines with an Application to Curve Design* Identities for trigonometric B-splines with an application to curve design. BIT 37, pp. 189-201.