# Performance Analysis for Methods for the Computation of Derivatives

Taeboom Jang* · Youngsong Cho* · Dongsoo Cho* · Deok-Soo Kim*

* Department of Industrial Engineering, Hanyang University

## Abstract

The computation of derivative of a rational Bézier curve is required on its various points for some applications. It can be an issue to quickly compute derivatives of rational Bézier curve. Therefore, some methods have been developed to compute derivatives of a rational Bézier curve.

This paper analyzes the performance of the methods for computing derivatives of a rational Bézier curve. Cons and pros of each method are also discussed. In this way, this paper discusses which method is preferable for the computation of derivatives of a rational Bézier curve in various situations.

### 1. Introduction

The first and higher derivatives of curves and surfaces provide the essential geometric information. The computation of derivatives is often required in various applications of curves and surfaces curve characterization [3], the detection of inflection points on a planar curve [3], the computation of tangent, normal and visibility cones [4] and so on. In some applications, it is necessary to compute derivatives of on various points of a given curve and surface. Therefore, it could be important to compute derivatives efficiently.

While non-uniform rational B-splines, which is usually denoted as NURBS, is widely used in representation and manipulation of geometry for various reasons, many theoretical researches deal with rational Bézier curves since the results may be directly applied to NURBS. Therefore, there exist several methods to compute derivatives of a rational Bézier curve. Sederberg and Wang found that the numerator of the derivative of a rational Bézier curve of degree $n$ can be represented as a Bézier curve of degree $2n-2$, and called a scaled hodograph [7]. Based on the work of [7], Sederberg presented the algorithms for evaluating points and tangents of a rational Bézier surface [8]. On the other hand, Floater expressed the derivative of a rational Bézier curve in terms of its control points and weights [1]. However, it has not been explicitly discussed which algorithm is preferable for the computation of derivatives of rational Bézier curve in different situations. This paper presents the evaluation of the performance of algorithms representing derivatives of rational Bézier curve. The performances are measured as computation requirements of derivatives of rational Bézier curve at the pre-processing stage and at the computation stage. Cons and pros of each method are also discussed. In this way, this paper discusses which method is preferable for the computation of derivatives of a rational Bézier curve in various situations.

## 2. Review

There exist several efforts to compute the derivative of a rational Bézier curve. In this section, these representations are reviewed.

### 2.1 Direct Differentiation (DD) Approach

This representation is a result of the direct differentiation of a rational Bézier curve without any manipulation. Despite this representations simplicity, it requires considerable operations of polynomials at the pre-processing stage and provides the insufficient geometric information. Furthermore, a serious numerical problem may occur due to its power form representation.

### 2.2 Scaled Hodograph (SH) Approach

A scaled hodograph, which is a numerator of derivative of a rational Bézier curve, is known to be a Bézier curve of degree $2n$-2 [7]. The name reflects the fact that a scaled hodograph provides the correct directions of derivatives of rational Bézier curve. However, a scaled hodograph may not provide the exact magnitude of derivative due to the loss of the information about the denominator. Therefore, scaled hodograph is useful in the situation that only the direction of the derivative is required.

### 2.3 Floaters Tangent (FT) Formula

Floater rewrote Equation [1] by the rearrangement of Bernstein polynomials in Reference [1] using Casteljaus algorithm for evaluating a point for evaluation of point on a Bézier curve. This representation provides good geometric intuition for the derivative of rational Bézier curve related with de Casteljaus algorithm for evaluating a point. Moreover, he provides the upper bound on the magnitude of the derivative.

### 2.4 Floaters Hodograph (FH) Formula

In Reference [1], Floater also proposed the formula for the derivative using new rational polynomial set with respect to weights. By introducing new polynomial set, he represented the derivative in terms of vectors joining consecutive control points. This represnetatiion has similar form of hodograph of Bezier curve. However, few properties of introduced polynomial set are kwon. Also, the formula is not fully investigate yet.

2.6 Closed Form Hodograph (CF) Approach

This approach means the representation suggested in this paper. It is useful to obtain arbitrary order of derivative of rational Bézier curve by recursive function call of the derivative routine.

## 3. Experiments and evaluation

The performances are measured as computation requirements of derivatives of rational Bézier curve. Experiments were performed to measure the computation requirements at the pre-processing stage and at the computation stage. These experiments can be contributed to determine which algorithm is preferable with respect to the frequency of the computation of the derivative. Before the discussion on the evaluation, readers should be noted that the experiments may be dependent on the actual implementations of the algorithms. Environment of experiments is shown in <Table 1>.

〈Table 1〉 Environment of experiments

| | |
|---|---|
| Platform | Pentium II 233 Mhz |
| OS | Windows NT 4.0 workstation |
| Programming Language | C++ |
| Compiler | Microsoft Visual C++ 4.0 |

For the evaluation of algorithms at the pre-processing stage, the computation time is measured that each algorithm spends for thirty randomly selected curves of 2,3, and 7 degree. And, two approaches are made for evaluation of the requirements at the computation stage. Honers rule and de Casteljaus algorithms are selected for the implementations of power basis and Bernstein basis, respectively. Besides the evaluation of computation time, analytic approach makes the differences among representations clear.

Major portion of computation requirement at the computation stage is spent for the computation of polynomials. Therefore, the performance to compute derivatives is sensitively affected by algorithms for computation of polynomials. While polynomials are expressed by power basis in direct differentiation and Floaters hodograph formula, other algorithms use Bernstein basis to represent polynomials. It is widely known that power basis is preferable to Bernstein basis for the efficiency of computation while ignoring computational errors. Therefore, conversion to power basis and Floaters hodograph formula are faster than other algorithms at the computation stage. If more efficient algorithm is selected for Bernstein basis, difference among representations will be diminished.

Based on the work of this thesis, the following observations could be deduced.

In situation that the computations of derivatives of rational Bézier curve are required in various points, the direct differentiation is preferable to other algorithms without considering the computation error. While it costs a considerable computation time at the pre-computation stage, the speed of direct differentiation is so fast for the computation of derivative.

Considering efficiency and computation error, Floaters tangent formula is proper at both the pre-processing stage and at the computation stage. Floaters tangent formula requires no computation at the pre-processing stage.

While Floaters tangent formula efficiently compute each derivative on curve, scaled hodograph provides overview for directions of derivatives on curve. Because, scaled hodograph represent direction of derivatives in a curve form.

While hodograph in closed form costs more computation requirements than other representation, this representation is useful to overview both magnitude and direction of derivative of rational Bézier curve. It is also useful to obtain arbitrary order of derivative of rational Bézier curve by recursive function call of the derivative routine. Moreover, the gap compared with other algorithms in performance at computation stage will be diminished by adoption of more efficient algorithm for Bernstein basis.

## 4. Conclusion

This paper presents new approach for computing derivatives of rational Bézier curve. This approach can relieve the computation of derivatives of rational Bézier curves and be extended to rational Bézier surface case. Also, other approaches are reviewed and their performances are evaluated.

## Reference


1. Floater, M. S., Derivatives of rational Bézier curves, *Computer Aided Geometric Design*, Vol. 9, pp.161-174, 1992.
2. Floater, M. S., Evaluation and properties of the derivative of a NURBS Curve, *Mathematical Methods in Computer Aided Geometric Design II*, pp. 261-274.
3. Kim, Deok-Soo, Hodograph approach to geometric characterization of parametric cubic curves., *Computer Aided Design*, Vol. 25, pp.644-654, 1993.
4. Kim, Deok-Soo,, Papalambros, P. Y. and Woo, T. C., Tangent, normal, and visibility cones on Bézier surfaces, *Computer Aided Geometric Design*, Vol. 12, pp.305-320, 1995.
5. Kim, Deok-Soo, Taeboom Jang and Young-Song Cho, The closed form of hodograph of rational Bézier curves and surfaces, *Transactions of the society of CAD/CAM engineers*, Vol. 3, No. 2, pp.135-139, 1998.
6. Saito T., Wang, Guo-Jin and Sederberg, T. W., Hodographs and normals of rational curves and surfaces, *Computer Aided Geometric Design*, Vol.12, pp.417-430, 1995.
7. Sederberg, T.W. and Wang, X., Rational hodographs, *Computer Aided Geometric Design*, Vol. 4, pp. 333-335, 1987.
8. Sederberg, T.W., Point and tangent computation of tensor product rational Bézier surfaces, *Computer Aided Geometric Design*, Vol. 12, pp. 103-106, 1995.