Maria Skublewska-Paszkowska[1], Edyta Łukasik[2], Jakub Smołka[3]

ANALYSIS OF MOTION INTERPOLATION METHODS

This article presents selected interpolation methods. The chosen algorithms were implemented in a computer program that was used for performing numerical tests. The results were analyzed in many aspects, including economic. The interpolation methods that were analyzed are: Catmull-Rom spline, the modified Catmull-Rom spline and blended parabolas. The methods were tested with respect to computation time and the accuracy of the interpolated data.

Keywords: *interpolation methods; numerical experiments; comparison; performance.*

IEL codes: *C63, C99.*

Марія Скублевска-Пашковска, Едита Лукашик, Якуб Смолка

АНАЛІЗ МЕТОДІВ ІНТЕРПОЛЯЦІЇ РУХУ

У статті розглянуто деякі методи інтерполяції руху. Вибрані алгоритми були введені в комп'ютерну програму, яка була використана для проведення чисельних експериментів. Отримані результати було проаналізовано в багатьох аспектах, у тому числі економічному. Проаналізовані методи інтерполяції включають сплайн Кетмулл-Рома, модифікований сплайн Кетмулл-Рома і метод змішаних парабол. Ці методи були протестовані з урахуванням часу обчислень і точності інтерпольованих даних.

Ключові слова: *методи інтерполяції; чисельні експерименти; порівняння; продуктивність.*

Мария Скублевска-Пашковска, Эдита Лукашик, Якуб Смолка

АНАЛИЗ МЕТОДОВ ИНТЕРПОЛЯЦИИ ДВИЖЕНИЯ

В данной статье рассмотрены некоторые методы интерполяции движения. Выбранные алгоритмы были введены в компьютерную программу, которая была использована для проведения численных экспериментов. Полученные результаты были проанализированы во многих аспектах, в том числе экономическом. Проанализированные методы интерполяции включают сплайн Кэтмулл-Рома, модифицированный сплайн Кэтмулл-Рома и метод смешанных парабол. Эти методы были протестированы с учетом времени вычислений и точности интерполированных данных.

Ключевые слова: *методы интерполяции; численные эксперименты; сравнение; производительность.*

Introduction. Interpolation is one of the key aspects in the motion animation. It provides fluency in the movement of the animated object. Interpolation concerns many of the animation parameters, such as: positions of the object in space, paths connecting those positions and object's orientation (angular position). Interpolation can be seen as one of the main foundations of animation.

Formally interpolation is a tool for computing intermediate values, between those from a given set of initial animation parameter values [7]. There are many various interpolation algorithms. They differ mainly in their complexity, time consumption and accuracy of the results [2].

[1] PhD, Professor Assistant, Institute of Computer Science, Faculty of Electrical Engineering and Computer Science, Lublin University of Technology, Poland.

[2] PhD, Professor Assistant, Institute of Computer Science, Faculty of Electrical Engineering and Computer Science, Lublin University of Technology, Poland.

[3] PhD, Professor Assistant, Institute of Computer Science, Faculty of Electrical Engineering and Computer Science, Lublin University of Technology, Poland.

Implementing interpolation in animation is not an easy task [5]. It involves (1) selecting a proper interpolation function, (2) parameterization of the function with respect to the motion path and (3) keeping control of speed at which the path is traversed (reparametrization of the interpolation function with respect to time).

This paper presents selected interpolation algorithms that can be applied to the interpolation of a position of a point in space. The paper analyzes both the use of the algorithms and their economic aspects.

Economic aspect of interpolation. As stated above, the selection and use of proper interpolation methods is a very important part of animation creation process. The economic aspect of the selected method should also be taken into consideration. There are algorithms which vary in: time consumption, complexity and quality of the produced results. Complexity of the algorithm is related to the time needed to compute interpolated values. The more time the interpolation takes, the more time is required to animate the scene. This increases the costs of creating animation. Lower complexity of the algorithm means that less computation is performed by computer processors. This saves time and money.

Another aspect that should be considered is the accuracy of interpolation. In order to achieve pleasing visual effects it is necessary to create fluent motion of an object. Moreover, there are restrictions on the object's positions in the scene that are being interpolated. It is impossible, or extremely difficult to interpolate between two locations that are far apart. Motion interpolation should give illusion of natural object movement.

A compromise between interpolation quality and interpolation cost needs be found. One has to find a low-cost, good interpolation algorithm.

Interpolation methods. There are various algorithms that allow interpolation of numerical data [9]. Selected methods are described below.

Linear interpolation. The simplest method is linear interpolation (equation 1). The interpolants (1-u and u) sum to one. This ensures that the interpolating curves fall within the convex hull of the geometric entities being interpolated [1].

$$P(u) = (1-u)P_0 + uP_1 \qquad (1)$$

It is possible to represent the equation 1 in more general form with the use of blending functions (F_0,F_1). This is shown in equation 2 [1]:

$$P(u) = F_0(u)P_0 + F_1(u)P_1 \qquad (2)$$

The same problem can be expressed in the terms of polynomial equations (equation 3) [1]:

$$P(u) = (P_1 - P_0)u + P_0 \qquad (3)$$

The algebraic form of the interpolation method is presented in equation 4 [1]:

$$P(u) = a_1u + a_0 \qquad (4)$$

Additionally, each of the above equations (1 to 4), can be shown in a matrix form. It is shown in equation 5, where the parameters are: U — variable, M — coefficients and B — geometric information respectively [1].

$$P(u) = [u \quad 1]\begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} P_0 \\ P_1 \end{bmatrix} = U^T MB \tag{5}$$

Matrix representation is very useful and the interpolation methods described in the following subsections are presented by means of this notation.

Hermite interpolation. In contrast to linear interpolation, Hermite method creates a cubic polynomial based on two separate points (e.g., P_i, P_{i+1}). It is also necessary to define tangent vectors at interpolation nodes (e.g., P'_i, P'_{i+1}). These vectors determine the shape of the final curve.

The Hermite matrices for the general representation shown in equation 5 are [1]:

$$U^T = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \tag{6}$$

$$M = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{7}$$

$$B = \begin{bmatrix} P_i \\ P_{i+1} \\ P'_i \\ P'_{i+1} \end{bmatrix} \tag{8}$$

A disadvantage of the Hermite interpolation is the fact that for each point (interpolation node) a tangent vector has to be defined. If there is a need to generate the curve based on large amount of points the tangent vectors become a real impediment. There are other methods where tangent vectors are computed automatically [1].

Catmull-Rom Spline. The Catmull-Rom spline (CM1/2) method creates a cubic polynomial, as in Hermite interpolation, but tangent vectors are generated automatically. For each point P_i tangent is computed by the method presented in equation 9 [1].

$$P'_i = \frac{1}{2}(P_{i+1} - P_{i-1}) \tag{9}$$

The tangent for the initial point is obtained from equation 10:

$$P'_0 = \frac{1}{2}(P_1 - (P_2 - P_1) - P_0) \tag{10}$$

Matrices U and B are the same as for Hermite interpolation (equation 6). Matrix M is shown in equation 11 [1].

$$M = \frac{1}{2}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \tag{11}$$

An internal tangent vector is independent of the position of middle point in the triple (P_{i-1},P_i, P_{i+1}) [1]. This is a drawback of this method. Its advantage, on the other hand, is the fact that computations are quick and simple.

Another method which allows obtaining Catmull-Rom spline (CMP) is to use relative position of the internal point (that is P_i) to independently determine the length of the tangent vector for each segment [8]. Thus, left and right hand tangents are given by equations (12, 13) [1]:

$$P'_{i-1}(1)=\frac{|P_i - P_{i-1}|}{|P_{i+1} - P_{i-1}|}(P_{i+1} - P_{i-1}) \tag{12}$$

$$P'_i(0)=\frac{|P_{i+1} - P_i|}{|P_{i+1} - P_{i-1}|}(P_{i+1} - P_{i-1}) \tag{13}$$

The cost of this method is only a little higher than the cost of the method given by equations (9) and (10).

Blended Parabolas. The construction of a blend of overlapping parabola segments is another way to define interpolation curves [1, 6]. The first and the last part of the curve are parabolas. Other parts are cubic segments (blend of two parabolas). A parabolic curve P(u) can be defined for 3 points using the following constrains: $P(0)=P_i$, $P(0.5)=P_{i+1}$ and $P(1)=P_{i+2}$. To the next group of 3 points P_{i+1}, P_{i+2}, P_{i+3} a parabola R(u) is fitted. Between P_{i+1} and P_{i+2} the two parabolas P(u) and R(u) overlap. The parabolas are linearly interpolated (blended) according to the equation (14) [1].

$$P(u)=(1-u)R(u)+uS(u) \tag{14}$$

This means that for the first two points and two ending points in the interpolated data set, parabolas are constructed. For each of the intermediate pairs of points a cubic curve is defined. The matrix M has the same coefficients as this matrix in Catmull-Rom spline (equation 11). This method assumes that all the points are equally spaced in parametric space.

Numerical experiments. A computer program that implemented the methods described above was created in order to compare their performance. It is a program which was prepared with the use of objective programming principles [10]. The most important aspect was the implementation of the vector and matrix operations (such as summation and multiplication). The overloading operators were used as a method of programming. The total of 312 tests were performed: 300 while testing the performance of interpolation methods and 12 in the accuracy assessment. Each test was repeated 10 times in order to compensate for variation in random data — the results were averaged over 10 repetitions.

The first aspect of the interpolation algorithms examined was the computation time for the following methods: Catmull-Rom spline (CM1/2), modification of Catmull-Rom spline (CMP) and blended parabolas (BP). The graphs of the resulting times are presented in Fig. 1, 2 and 3. The axis of ordinates corresponds to the number of interpolation points while the other axis corresponds to the computation time. Each graph presents the results obtained for the interpolation steps (ΔU) which are respectively 0.1, 0.5 and 0.01.

There are two important conclusions that can be drawn from the obtained results. The first concerns the way the computation time changes — it depends linearly on the number of interpolation points. Moreover, the time which is needed for computing tangents is also linear. The second conclusion is that the results show that the blended parabolas method is the quickest from all of the tested. This is due to the fact that both Catmull-Rom methods require tangents computations.

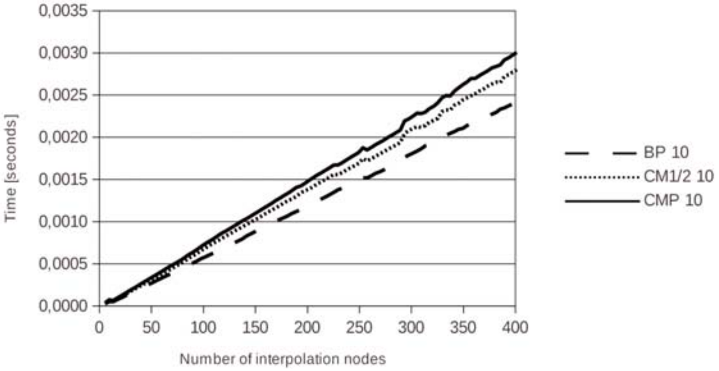

Figure 1. **Time computation for the interpolation ($\Delta U=0.1$)**

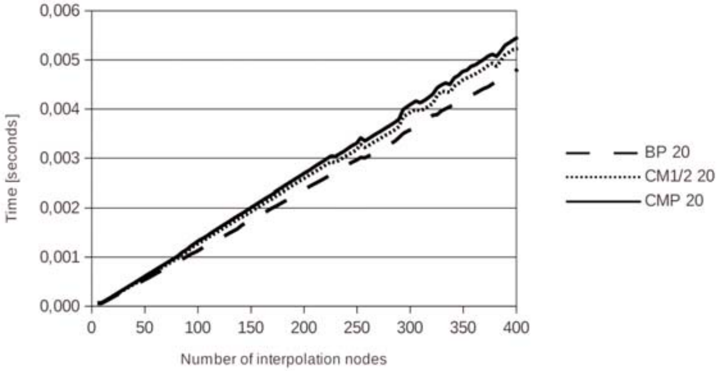

Figure 2. **Time computation for the interpolation ($\Delta U=0.5$)**

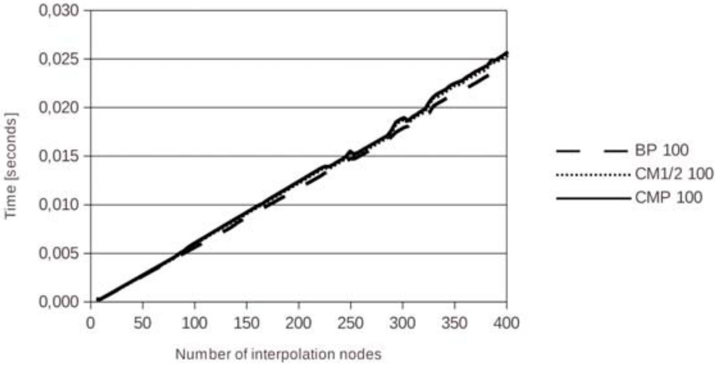

Figure 3. **Time computation for the interpolation ($\Delta U=0.1$)**

The second aspect of the interpolation algorithms that was analyzed was the accuracy of the interpolated data. This test was based on the following methods: Catmull-Rom spline (CM1/2), modification of Catmull-Rom spline (CMP) and blended parabolas (BP). Random sets of 401 points were generated. Then, in consecutive tests, every 4th, 8th, 16th point was copied into the data set that was later used for interpolation. The rest of the points were left out. The results of data interpolation are presented in Fig. 4, 5 and 6.

Once the interpolated points were obtained, the distance between the original data set consisting of 401 points and the interpolated output was computed. The difference was computed in several steps. First, for each output point the nearest input point on the ordinate axis was selected. Second, the difference of the y coordinates of these points was computed. Third, the absolute values of these differences were summed. The results are presented in Table 1.

Table 1. **The average distance between original and interpolated data**

# of po-ints	BP			CM1/2			CMP		
	Δ U=0.1	Δ U=0.2	Δ U=0.01	Δ U=0.1	Δ U=0.2	Δ U=0.01	Δ U=0.1	Δ U=0.2	Δ U=0.01
100	287,4846	287,5248	287,142	779,693	833,4784	827,051	806,183	839,8394	843,1418
50	419,9992	416,0748	414,4772	970,9398	943,1184	938,292	981,1458	964,3258	950,9866
25	606,2346	597,5464	593,3212	1103,7288	1063,3986	1007,2592	1079,7496	1042,2162	1017,312

The first conclusion is that the interpolation step has little influence on the accuracy measure. Additionally, the more input points, the lower the difference. This is caused by the density of the interpolated points. The blended parabolas (BP) method offers the highest level of accuracy.

The Fig. 4 presents interpolated curves generated for 100 input points. The figure shows only a part of each curve (the first 20 points) for improved readability. The data presented in Table 1 correspond to the curves in Fig. 4. Fig. 5 and 6 present interpolated curves generated respectively for 50 and 25 input points.

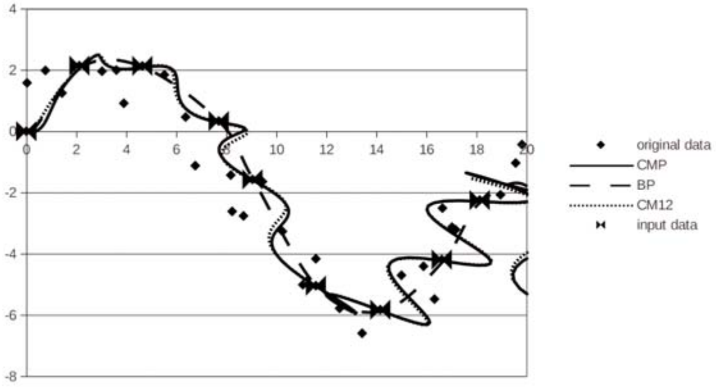

Figure 4. **The accuracy comparison for the selected methods for selected points (every 4th from initial set)**

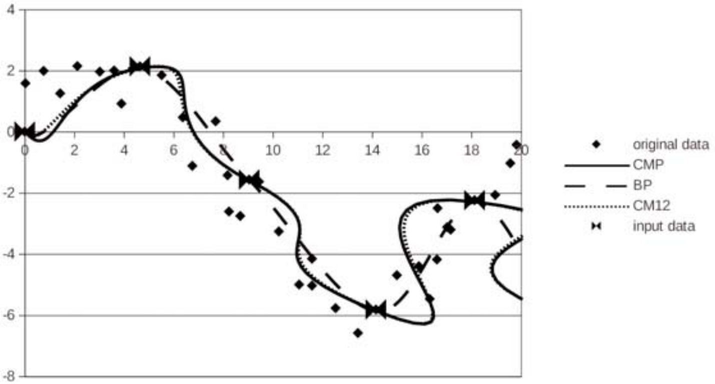

Figure 5. **The accuracy comparison of the selected methods for the selected points (every 8th from the initial set)**

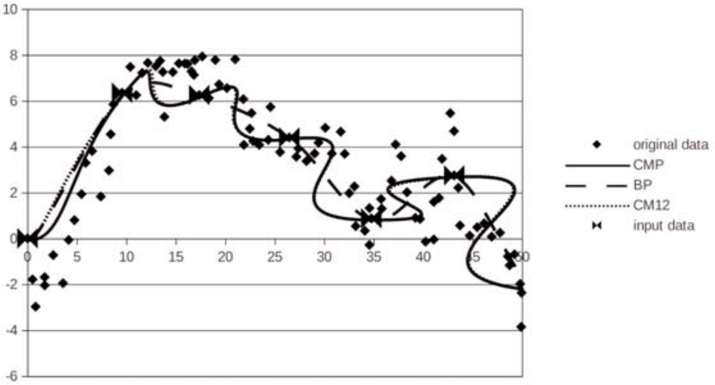

Figure 6. **The accuracy comparison of the selected methods for the selected points (every 16th the from initial set)**

All of the performed tests clearly indicate that the blended parabolas method is the most effective with respect to accuracy.

Conclusion. The article describes selected interpolation methods. They can be used for various purposes. They can be applied to motion capture analysis [3] but also to economic data. Interpolation methods can be used in many different computing environments including cloud computing [4].

3 main interpolating methods were compared: Catmull-Rom spline, modification of Catmull-Rom spline and blended parabolas. Both the computation time and the accuracy of the interpolated data were analyzed. The numerical experiments were performed for various numbers of points (interpolation nodes) and interpolation steps (value of the *u* parameter belongs to the interval of (0, 1]).

Blended parabolas turned out to be the best method according to the performed tests. The best performance is caused by the fact that this method does not require computation of tangent vectors. The other advantage of this method is the lack of the significant influence of interpolation step on the accuracy.

Refrences:

Parent, R. (2008). Computer Animation: Algorithms & Techniques, Elsevier.

Engeln-Mullges, G., Uhlig, F. (1996). Numerical Algorithms with C, Springer.

Koziel, G., Laskowski, M. (2012). Motion capture, Actual Problems of Economy, 12:376-380.

Badurowicz, M., Szymczyk, T. (2012). Economic aspects of cloud computing, Actual Problems of Economy.

Yamane, K. (2004). Simulating and Generatins Motions of Human Figures, Springer Tracts in Advanced Robotics 9, Springer-Verlag.

Barger, R. (1992). Trajectory fitting in Function Space With Application to Analytic Modeling of Surfaces, NASA, Technical Paper 3232.

Rogers, D., Adams, J. (1990). Mathematical Elements for Computer Graphics, McGraw-Hill.

Rovenski, V. (2000). Geometry of Curves ans Surfaces with Maple, BirkHauser, Boston.

Salomon, D. (1999). Computer Graphics and Geometric Modeling, Springer-Verlag New York.

Eckel, B. (2002). Thinking in C++", Helion.

Стаття надійшла до редакції 22.07.2012