

# PORÓWNANIE WYBRANYCH METOD INTERPOLACJI RUCHU

**Jakub Smolka, Maria Skublewska-Paszowska**

Politechnika Lubelska, Wydział Elektrotechniki i Informatyki

**Streszczenie.** Interpolacja jest jednym z kluczowych elementów wykorzystywanych w animacji komputerowej. Dobór odpowiedniej metody interpolacji wpływa na ruch animowanej postaci. Artykuł przedstawia wybrane metody interpolacji i porównuje je ze względu na czas wykonywania obliczeń oraz dokładność uzyskanych wyników. Algorytmy, które przeanalizowano to: metoda Catmulla-Roma, zmodyfikowana metoda Catmulla-Roma oraz krzywe przejściowe między parabolami (blended parabolas). Eksperymenty numeryczne przeprowadzono za pomocą programu komputerowego napisanego w języku C++.

**Słowa kluczowe:** metody interpolacji, Hermite, krzywe sklepane Catmulla-Roma, krzywe sklepane Blended Parabolas

## COMPARISON OF THE SELECTED MOTION INTERPOLATION METHODS

**Abstract.** Interpolation is one of key aspects of computer animation. The selection of the proper interpolation method influences motion of animated objects. The paper presents selected interpolation methods and compares them with respect to computation time and accuracy. The three algorithms that were analyzed are: Catmull-Rom Spline, modified Catmull-Rom Spline and Blended Parabolas method. Numerical experiments were performed using a program written in C++ language.

**Keywords:** interpolation methods, Hermite, Catmull-Rom, Blended Parabolas

### Wstęp

Animowanie ruchu obiektu polega na zmienianiu wartości parametrów, które wpływają na jego położenie. Ze względu na to, że dane ruchu nie są dostępne z wymaganą dokładnością, jednym z kluczowych elementów animacji obiektu jest dobór odpowiedniej metody interpolacji. Interpolacja może być zastosowana do wyznaczania wartości wielu parametrów animacji takich jak: pozycja obiektu, czy jego orientacja (pozycja kątowa) w przestrzeni. Dlatego można stwierdzić, że interpolacja jest jednym z fundamentów animacji komputerowej. Algorytmy interpolacji pozwalają na przedstawienie płynnego ruchu animowanego obiektu w sposób zbliżony do naturalnego.

Interpolacja jest narzędziem, dzięki któremu można wyznaczyć wartości pośrednie, pomiędzy danymi należącymi do zbioru zadanych parametrów animacji [5]. Istnieje wiele różnych algorytmów interpolacji, które różnią się złożonością, czasem działania oraz dokładnością wyników [3].

Zastosowanie interpolacji w animacji nie jest łatwym zadaniem [9]. W celu jej użycia należy wykonać kolejne etapy jak: (1) wybór odpowiedniej funkcji interpolacji, (2) parametryzację funkcji uwzględniając ścieżkę ruchu obiektu, oraz (3) zapewnienie kontroli nad położeniem obiektu (punktu) w zależności od czasu (ustalenie zmian prędkości – ponowna parametryzacja funkcji interpolacyjnej w czasie).

Artykuł ten przedstawia wybrane algorytmy interpolacji ruchu, które mogą zostać zastosowane do interpolacji pozycji punktu w przestrzeni. Ponadto zawiera on analizę wybranych algorytmów interpolacyjnych przeprowadzoną na podstawie opisanych testów związanych ze złożonością i dokładnością badanych metod.

### 1. Wybrane metody interpolacji

Istnieje wiele różnych algorytmów, które pozwalają na interpolację danych liczbowych [7]. Krzywe stosowane do budowy funkcji interpolacyjnej mogą być różne. Można je podzielić na [4]: (1) wielomianowe, (2) liniowe, które są szczególnym przypadkiem wielomianowych, (3) wymierne oraz (4) przestępne. Wybrane algorytmy zostały przedstawione poniżej.

#### Interpolacja liniowa

Najprostszą metodą interpolacji jest interpolacja liniowa, której równanie zostało przedstawione wzorem 1. Wyrażenia:  $1-u$  oraz  $u$  sumują się do jedności. Gwarantuje to, że otrzymana krzywa interpolacyjna (w tym przypadku odcinek) zawarta jest w otocze wypukłej danych punktów [4, 8].

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

Liniową interpolację (wzór 1) można przedstawić w bardziej ogólnej postaci przy pomocy funkcji mieszających (ang. *blending functions*) –  $F_0, F_1$ . Reprezentacja ta została przedstawiona wzorem 2 [4, 8].

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

W przypadku tej reprezentacji muszą być spełnione warunki gwarantujące budowę odcinka:  $F_0(u) + F_1(u) = 1$  oraz  $0 \leq F_0(u) \leq 1$ .

Ten sam problem może zostać rozwiązany przy pomocy równania wielomianowego (wzór 3) [4, 8].

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

Algebraiczna forma tego równania została przedstawiona we wzorze 4 [4, 8].

$$P(u) = a_1 u + a_0 \quad (4)$$

Dodatkowo, każde z zaprezentowanych równań (od 1 do 4) można przedstawić w postaci macierzowej. Forma ta została pokazana wzorem 5 [4].

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

gdzie:  $U$  - macierz zawierająca parametry,  $M$  - macierz zawierająca ustalone współczynniki,  $B$  - macierz zawierająca informacje geometryczne.

Ze względu na fakt, że reprezentacja macierzowa jest bardzo wygodna, a także łatwa w implementacji, wszystkie kolejne metody interpolacji w tym rozdziale zostaną przedstawione w tej formie.

#### Interpolacja Hermite'a

W przeciwieństwie do interpolacji liniowej, algorytm Hermite'a tworzy krzywą trzeciego stopnia pomiędzy dwoma punktami (przykładowo  $P_i, P_{i+1}$ ). W tej metodzie niezbędne jest utworzenie wektorów pochodnych parametryzacji ( $p'_i, p'_{i+1}$ ), które mają wpływ na kształt tworzonej krzywej – wektory są styczne do krzywej w punktach  $P_i, P_{i+1}$  [4, 8].

Parametry równania macierzowego (ze wzoru 5) zostały podane we wzorach od 6-8.

$$U^T = [u^3 \ u^2 \ u \ 1] \quad (6)$$

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

$$B = \begin{bmatrix} P_i \\ P_{i+1} \\ p_i \\ P_{i+1} \end{bmatrix} \quad (8)$$

Dużym utrudnieniem podczas stosowania tej metody jest konieczność wyznaczenia wektorów pochodnych parametryzacji dla każdego węzła interpolacyjnego ( $p_i, p_{i+1}$ ). W przypadku, gdy algorytm Hermite'a zastosowany jest do większej liczby punktów, definiowanie wektorów pochodnych staje się znaczącą przeszkodą. Istnieją metody interpolacji, w których wektory te generowane są automatycznie [4].

### Krzywe sklepane Catmulla-Roma

Podobnie, jak w przypadku interpolacji Hermite'a, stosowanie krzywych sklepanych tworzy łuki trzeciego stopnia pomiędzy kolejnymi parami punktów. Krzywe te można rozpatrywać jako powstałe z wielomianowych łuków Hermite'a. Pochodne w punktach złączenia krzywych są łatwe do obliczenia, ze względu na ciągłość pochodnej drugiego rzędu. Zaletą tej metody jest automatyczne wyznaczenie wektorów pochodnych. Dla każdego punktu  $P_i$ , będącego punktem łączącym łuki, wektor jest generowany zgodnie ze wzorem 9 (jest on równy połowie różnicy punktu poprzedzającego oraz wcześniejszego) [4, 8].

$$p_i = \frac{1}{2}(P_{i+1} - P_{i-1}) \quad (9)$$

Wektor pochodnych dla punktu początkowego ( $P_0$ ) i końcowego może być wyznaczany w różny sposób (np. poprzez podanie ich przez użytkownika lub ich automatyczny dobór). Jedną z metod jego obliczenia przedstawia wzór 10 [4, 8].

$$p_0 = \frac{1}{2}(P_1 - (P_2 - P_1) - P_0) \quad (10)$$

Przedstawiając algorytm krzywych sklepanych Catmulla-Roma w postaci macierzowej, należy zastosować macierz  $U$  identyczną z podaną dla metody Hermite'a (wzór 6), natomiast macierze  $M$  oraz  $B$  dane są odpowiednio wzorami 11 oraz 12 [4, 8]:

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

$$B = \begin{bmatrix} P_{i-1} \\ P_i \\ p_i \\ P_{i+2} \end{bmatrix} \quad (12)$$

Metoda ta w dalszej części artykułu będzie nazywana krótko: CM1/2.

Największą zaletą tej metody jest jej prostota oraz szybkość obliczeń. Główna niedogodność stosowania krzywych sklepanych Catmulla-Roma polega na tym, że wektor pochodnej punktu  $P_i$  łączącego dwie krzywe nie zależy od położenia tego punktu względem punktów  $P_{i-1}$  i  $P_{i+1}$  [4]. Dlatego istnieją inne konstrukcje wyznaczania wektorów pochodnych bazujące na tej metodzie (np. przyjęcie wektora prostopadłego do dwusiecznej kąta między odcinkami) [4].

Zmodyfikowana metoda Catmulla-Roma pozwala na niezależne dobranie długości wektora pochodnej w punkcie ( $P_i$ ), w którym się łączą dwa łuki [4, 6]. Wtedy lewy i prawy wektor pochodnej łuku wyznaczany jest zgodnie ze wzorami 13 (pomiędzy punktami  $P_{i-1}$  oraz  $P_i$ ) i 14 (pomiędzy punktami  $P_i$  oraz  $P_{i+1}$ ) [4, 8].

$$p_i' = \frac{|P_i - P_{i-1}|}{\|P_{i+1} - P_{i-1}\|} (P_{i+1} - P_{i-1}) \quad (13)$$

$$p_i' = \frac{|P_{i+1} - P_i|}{\|P_{i+1} - P_{i-1}\|} (P_{i+1} - P_{i-1}) \quad (14)$$

Koszt wyznaczenia wektorów jest trochę większy niż tych otrzymanych przy pomocy wzorów 9 oraz 10 [4]. Metoda ta w dalszej części artykułu będzie nosiła skróconą nazwę CMP.

### Krzywe przejściowe między parabolami

Wyznaczanie krzywych przejściowych pomiędzy parabolami jest kolejną metodą uzyskiwania krzywych interpolacyjnych dla punktów, które tworzą łuki wielomianowe trzeciego stopnia [1, 4]. Dwie części krzywej - pierwsza i ostatnia są parabolami. Każdy z pozostałych łuków wyznaczany jest na podstawie czterech punktów i stanowi segment trzeciego stopnia otrzymany z interpolacji dwóch parabol. Powstają one w następujący sposób - parabola  $R(u)$  jest określona przez trzy kolejne punkty przy nałożeniu ograniczeń:  $R(0)=P_i$ ,  $R(0.5)=P_{i+1}$  oraz  $R(1)=P_{i+2}$ , a parabola  $S(u)$  przy analogicznych ograniczeniach dotyczących punktów:  $P_{i+1}$ ,  $P_{i+2}$ ,  $P_{i+3}$ . Pomiędzy punktami  $P_{i+1}$  oraz  $P_{i+2}$  dwie parabole  $R(u)$  oraz  $S(u)$  zachodzą na siebie. Parabole te są interpolowane liniowo, co przedstawia wzór 15 [4, 8].

$$P(u) = (1-u)R(u) + uS(u) \quad (15)$$

Macierz  $M$  z postaci macierzowej dla tej metody została przedstawiona wzorem 16 [4].

$$M = \frac{1}{2} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \quad (16)$$

## 2. Eksperymenty numeryczne

Ze względu na fakt, że najczęściej w grafice komputerowej stosowane są wielomianowe krzywe trzeciego stopnia, zwane także kubicznymi, przedstawione testy dotyczą tej grupy funkcji interpolacyjnych.

Porównywane algorytmy zostały zaimplementowane przez autorów przy wykorzystaniu języka C++ [2]. Stworzony program pozwala na interpolację danych wejściowych i na generowanie danych wyjściowych interpolowanych z dowolną dokładnością. Kolejne metody interpolacyjne zostały zaimplementowane przy użyciu postaci macierzowej. Najważniejsze operacje, które umożliwiły obliczanie interpolowanych danych to dodawanie oraz mnożenie macierzy (także wektorów). Metody zostały zaimplementowane przy pomocy techniki przeciążania operatorów.

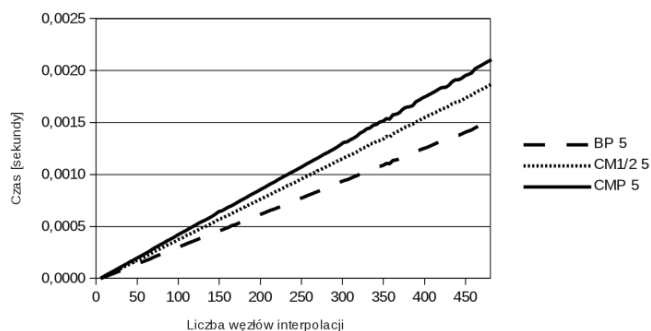
Przedstawiona poniżej analiza dotyczy dwóch aspektów interpolacji: czasu obliczeń wybranych algorytmów oraz dokładności interpolowanych danych.

Próbie czasu wykonano na 4800 zestawach danych, natomiast próbie dokładności przeprowadzono dla 80 zestawów danych wejściowych. Dane generowano losowo. W pierwszym wypadku maksymalna liczba węzłów interpolacyjnych wynosiła 480, zaś w drugim została ustawiona na 481 we wszystkich testach.

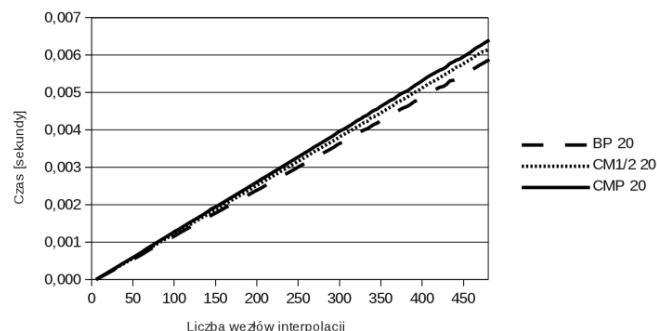
### Analiza czasu obliczeń wybranych algorytmów interpolacji

Jak już wspomniano, pierwsze z analizowanych zagadnień, dotyczy szybkości wykonywania obliczeń dla wybranych metod interpolacyjnych. Przetestowano opisane wcześniej metody: krzywych sklepanych Catmulla-Roma (CM1/2), zmodyfikowanej metody Catmulla-Roma (CMP) oraz krzywych przejściowych pomiędzy parabolami (BP). Wykresy przedstawiające czasy obliczeń dla analizowanych algorytmów przedstawiają rysunki 1 oraz 2.

Oś rzędnych reprezentuje liczbę węzłów interpolacji, podczas gdy oś odciętych przedstawia czas wykonywania poszczególnych metod. Każdy z rysunków ilustruje wyniki uzyskane dla kroku interpolacji ( $\Delta U$ ), odpowiednio: 0,2 dla pierwszego przykładu oraz 0,05 dla drugiego.



Rys. 1. Czas obliczeń dla kroku interpolacji równego 0,2



Rys. 2. Czas obliczeń dla kroku interpolacji równego 0,05

Z wyników przedstawionych na rysunkach 1 i 2 można wysnuć dwa wnioski. Pierwszy związany jest z czasem obliczeń, który jest zróżnicowany. Zależność ta jest liniowa w stosunku do liczby węzłów interpolacji. Dodatkowo, czas potrzebny na wyznaczenie wektorów pochodnych łączących dwie krzywe, jest także liniowo zależny od liczby węzłów. Przedstawione wyniki jednoznacznie wskazują, że najszybszą metodą z analizowanych jest algorytm wyznaczania krzywych sklejanych pomiędzy parabolami (oznaczona jako BP). Wynika to głównie z faktu, że w pozostałych metodach należy wyznaczać wektory pochodnych łączących dwie krzywe.

### Analiza dokładności interpolowanych danych wybranych algorytmów

Drugim analizowanym zagadnieniem była dokładność interpolowanych danych. Tak jak w przypadku próby czasu porównano następujące algorytmy: krzywych sklejanych Catmulla-Roma (CM1/2), zmodyfikowanej metody Catmulla-Roma (CMP) oraz krzywych przejściowych pomiędzy parabolami (BP).

Początkowy zbiór danych wejściowych składał się z 481 elementów. Testy, które zostały przeprowadzone i przedstawione w tym artykule, były wykonywane na danych wybranych ze zbioru wejściowego. Wybierane były węzły, odpowiednio co drugi i co czwarty. Dzięki temu sprawdzono jak bardzo można obniżyć szczegółowość danych wejściowych bez znacznego pogorszenia wyników interpolacji. Przykładowe wyniki pokazano na rysunkach 3, 4, 5 oraz 6.

Po otrzymaniu interpolowanych punktów, należało obliczyć, jak dokładna okazała się analizowana metoda interpolacyjna. W tym celu obliczana była odległość pomiędzy punktami wejściowymi, a odpowiadającymi im punktami wyjściowymi. Różnica ta była obliczana w dwóch etapach. Po pierwsze, dla każdego punktu interpolowanego wyznaczany był najbliższy punkt z danych wejściowych (według odległości osi rzędnych). Następnie wyznaczana była odległość (euklidesowa) tych punktów. Obliczone odległości zsumowano. Otrzymane wyniki zostały przedstawione w tabelach 1, 2 oraz 3, dla kolejno analizowanych algorytmów interpolacyjnych.

Wnioski, które nasuwają się po przeanalizowaniu danych w tabelach 1-3, są następujące: (1) dwukrotne zmniejszenie liczby punktów w zbiorze danych wejściowych powoduje zauważalne zmniejszenie dokładności otrzymanych wyników, otrzymany błąd nie jest jednak, w większości przypadków, dwukrotnie wyższy,

(2) zmniejszenie kroku interpolacji powoduje poprawę dokładności, jednak czterokrotna zmiana tego kroku nie powoduje czterokrotnego zmniejszenia błędu, (3) najlepszą metodą interpolacji okazał się algorytm wyznaczania krzywych sklejanych pomiędzy parabolami – błędy interpolacji były najmniejsze, a ponadto zmiana kroku interpolacji ma niewielki wpływ na jej dokładność.

Rysunki od 3 do 6 zawierają wyniki dokładności uzyskanych krzywych interpolacyjnych dla wybranych metod względem punktów wejściowych. Na rysunkach zaznaczono punkty oryginalnego zestawu danych oraz punkty wybrane jako dane wejściowe dla algorytmów (co drugi lub co czwarty) ze zbioru początkowego). Na podstawie wybranych punktów przeprowadzono interpolację dla dwóch liczb węzłów interpolacyjnych oraz dla dwóch wartości parametru  $\Delta U$  0,2 oraz 0,05.

Tabela 1. Obliczona odległość między punktami wejściowego i interpolowanych dla metody krzywych sklejanych Catmulla-Roma (CM1/2)

Liczba punktów w zbiorze wejściowym	CM1/2	
	$\Delta U=0,2$	$\Delta U=0,05$
243	896,1	675,9
123	1421,7	993,6

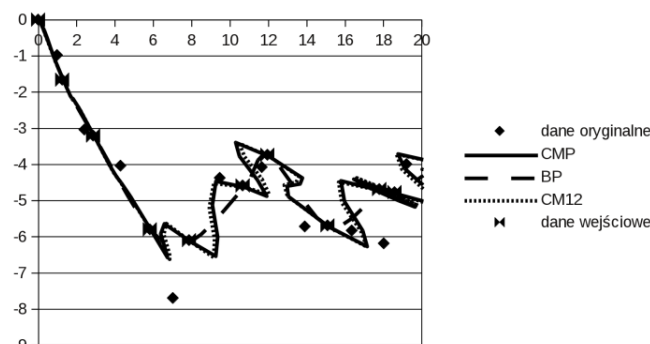
Tabela 2. Obliczona odległość między punktami wejściowego i interpolowanych dla metody krzywych sklejanych Catmulla-Roma (CMP)

Liczba punktów w zbiorze wejściowym	CMP	
	$\Delta U=0,2$	$\Delta U=0,05$
243	943,2	714,6
123	1394,1	992,1

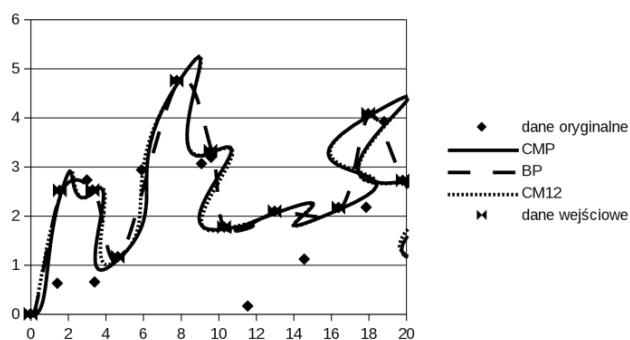
Tabela 3. Obliczona odległość między punktami wejściowego i interpolowanych dla metody krzywych sklejanych pomiędzy parabolami (BP)

Liczba punktów w zbiorze wejściowym	BP	
	$\Delta U=0,2$	$\Delta U=0,05$
243	182,9	189,4
123	357,3	328,8

Rysunek 3 przedstawia interpolowane krzywe uzyskane analizowanymi metodami dla zbioru wejściowego o liczebności 243 elementów. Dla przejrzystości przedstawiono tylko pierwszych 20 elementów. Punkty te zostały uzyskano ze zbioru początkowego, wybierając co drugi element. Krok interpolacji został ustalony na 0,2.

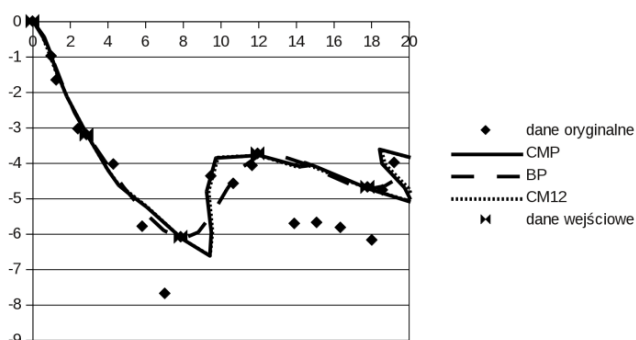
Rys. 3. Porównanie dokładności wybranych metod interpolacji dla pierwszych 20 punktów (co drugi ze zbioru początkowego) i  $\Delta U=0,2$ 

Rysunek 4 przedstawia interpolowane krzywe uzyskane analizowanymi metodami dla zbioru wejściowego o liczebności 243 elementów. Tutaj również dla przejrzystości przedstawiono tylko pierwszych 20 elementów. Punkty te uzyskano ze zbioru początkowego, wybierając co drugi element. Krok interpolacji był większy i został ustalony na 0,05.



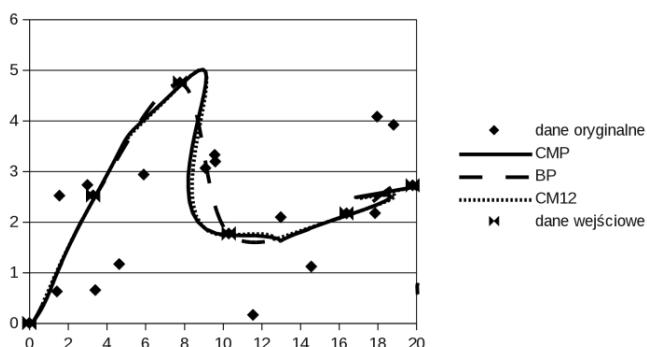
Rys. 4. Porównanie dokładności wybranych metod interpolacji dla pierwszych 20 punktów (co drugi ze zbioru początkowego) i  $AU=0,05$

Rysunek 5 przedstawia wynik kolejnej próby - krzywe uzyskane w wyniku interpolacji dla zbioru wejściowego o liczebności 123 elementów. Tak jak poprzednio przedstawiono tylko pierwszych 20 punktów. Punkty te zostały uzyskano ze zbioru początkowego, wybierając co czwarty element. Krok interpolacji  $\Delta U$  został ustalony na 0,2.



Rys. 5. Porównanie dokładności wybranych metod interpolacji dla 20 punktów (co czwarty ze zbioru początkowego) i  $AU=0,2$

Rysunek 6 przedstawia interpolowane krzywe uzyskane analizowanymi metodami dla zbioru wejściowego o liczebności 123 elementów. Punkty te zostały uzyskano ze zbioru początkowego, wybierając co czwarty element. Krok interpolacji wynosił 0,05.



Rys. 6. Porównanie dokładności wybranych metod interpolacji dla 20 punktów (co czwarty ze zbioru początkowego) i  $AU=0,05$

Również wyniki przedstawione na rysunkach wskazują, że najlepszą metodą interpolowania krzywych okazała się metoda krzywych przejściowych pomiędzy parabolami.

### 3. Podsumowanie

W artykule omówiono wybrane metody interpolacji danych, które mogą zostać zaimplementowane w różnych programach komputerowych. Mogą one zostać zastosowane między innymi podczas analizy czy wizualizacji danych ruchu, otrzymanych z systemów akwizycji ruchu (ang. Motion Capture Systems).

W artykule tym skupiono uwagę na trzech algorytmach interpolacji danych: metodzie krzywych sklejanych Catmulla-Roma, zmodyfikowanej metodzie Catmulla-Roma oraz metodzie krzywych przejściowych pomiędzy parabolami. Przeanalizowano zarówno szybkość obliczeń, a także dokładność danych otrzymanych w wyniku interpolacji. Wykonano eksperymenty numeryczne dla różnej liczby węzłów interpolacyjnych oraz kroku interpolacji.

Przeprowadzone testy pozwalają stwierdzić, że ze względu na przyjęte kryteria, najlepszą metodą okazała się metoda krzywych przejściowych pomiędzy parabolami. Jednym z powodów jest fakt, że nie wymaga ona generowania wektorów pochodnych. Dodatkową zaletą tego algorytmu jest niewielki wpływ kroku interpolacyjnego na dokładność uzyskiwanych wyników.

### Literatura

- [1] Barger R.: Trajectory fitting in Function Space With Application to Analytic modeling of Surfaces, NASA, Technical Paper 3232, 1992.
- [2] Eckel B.: Thinking in C++, Helion 2002.
- [3] Engel-Mullges G., Uhlig F.: Numerical Algorithms with C, Springer 1996.
- [4] Parent R.: Computer Animation: Algorithms & Techniques, Elsevier 2008.
- [5] Rogers D., Adams J.: Mathematical Elements for Computer Graphics, McGraw-Hill, 1990.
- [6] Rovenski V.: Geometry of Curves and Surfaces with Maple, BirkHauser, Boston, 2000.
- [7] Salomon D.: Computer Graphics and Geometric Modeling, Springer-Verlag New York, 1999.
- [8] Skublewska-Paszowska M., Łukasik E., Smolka J.: Analysis of motion interpolation methods, Shatsk Lakes, Ukraine, July 2-6, 2012, in press.
- [9] Yamane K.: Simulating and Generating Motions of Human Figures, Springer Tracts in Advanced Robotics 9, Springer-Verlag 2004.

**Dr inż. Maria Skublewska-Paszowska**  
e-mail: maria.paszowska@pollub.pl

Pracownik naukowo-dydaktyczny pracujący w Instytucie Informatyki Wydziału Elektrotechniki i Informatyki Politechniki Lubelskiej. Tytuł magistra uzyskała na Politechnice Lubelskiej. Stopień doktora otrzymała na Politechnice Śląskiej. Działalność naukowa związana jest z: transformacjami falkowymi, miarami jakości oceny wizualnej obrazów i zastosowania ich w adaptacyjnej kompresji obrazów, a także analizą ruchu.



**Dr inż. Jakub Smolka**  
e-mail: jakub.smolka@pollub.pl

Pracownik naukowo-dydaktyczny w Instytucie Informatyki Wydziału Elektrotechniki i Informatyki Politechniki Lubelskiej. Tytuł magistra uzyskał na Politechnice Lubelskiej natomiast stopień doktora na Politechnice Śląskiej. Jego działalność naukowa związana jest z przetwarzaniem obrazów cyfrowych w szczególności z ich segmentacją oraz kompresją, a także analizą ruchu.



otrzymano/received: 19.11.2012

przyjęto do druku/accepted: 23.05.2013