

Metody wyznaczania wskaźników podobieństwa ruchu trójwymiarowego

Piotr Flisiak*, Marcin Kuszyk

Politechnika Lubelska, Katedra Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. Praca zawiera sposoby porównywania ruchów trójwymiarowych, zapisanych w pliku C3D za pomocą korelacji liniowej Pearsona i błędu średnio-kwadratowego. Została zbadana także ich dokładność. Ruchy osoby ćwiczącej na ergometrze na dystansie 500 metrów zostały zarejestrowane technologią akwizycji ruchu. W artykule została przedstawiona utworzona aplikacja, użyta do analizy.

Słowa kluczowe: Porównywanie ruchu; Ruch trójwymiarowy; Algorytmy podobieństwa ruchu.

*Autor do korespondencji.

Adres e-mail: piotr.flisiak.it@gmail.com

Methods of determining indicators of similarity of 3D motion

Piotr Flisiak*, Marcin Kuszyk

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract. This paper includes methods for comparing three-dimensional movements with data saved in a C3D file using Pearson's linear correlation algorithms and a mean-quadratic error. Their accuracy was also examined. The movements of a person rowing on an ergometer at a distance of 500 meters were recorded by motion acquisition technology. The article presents the created application used for analysis.

Keywords: Comparing moves; Three-dimensional movement; Movement similarity algorithms.

*Corresponding author.

E-mail address: piotr.flisiak.it@gmail.com

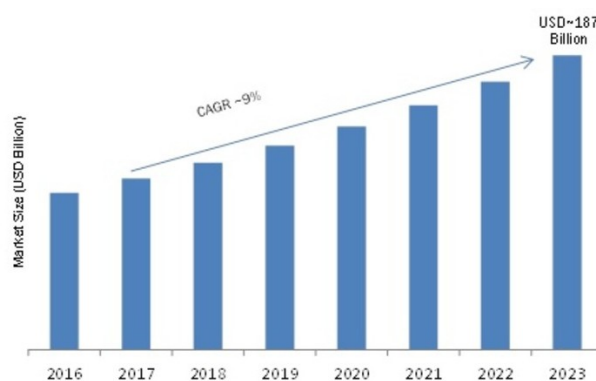
1. Wstęp

Ruch trójwymiarowy jest podstawą w życiu człowieka, dlatego też od dawna ludzie pracowali nad jego jak najbardziej dokładnym opisem, aby móc zrozumieć jego charakterystykę i jak najlepiej go wykorzystać. Obecnie w świecie komputerowym coraz dokładniej symulowany jest ten ruch wykorzystując znane światu prawa fizyki i nowe technologie, jedną z nich jest metoda przechwytywania ruchu motioncapture. Motion capture (mocap) to próbkowanie i rejestrowanie ruchu ludzi, zwierząt i nieożywionych obiektów jako dane 3D. Dane mogą być wykorzystane do badania ruchu lub do animowania życia modeli komputerowych 3D. Ponieważ większość aplikacji mocap wymaga obecnie specjalnego sprzętu, nadal istnieje ograniczona liczba firm, szkół i organizacji korzystających z technologii mocap [1]. Zostaje jednak pytanie jak sprawdzić czy dany ruch jest prawidłowy i powtarzalny. W dalszej części artykułu zostanie przedstawiona odpowiedź na to pytanie. W tym celu napisano aplikację analizującą dwa algorytmy porównywania ruchu na plikach C3D.

2. Zastosowanie i rynek

Na początku należy określić przeznaczenie danego algorytmu, gdyż nie do każdego celu będzie pasował dany algorytm porównujący ruch. W medycynie na ten przykład mniej będzie ważna sama pozycja danego znacznika a bardziej kąt pomiędzy nimi aby sprawdzić np. zakres ruchu [2]. W sporcie natomiast korzysta się będzie zarówno z kąta, jak i z samej pozycji znacznika ponieważ ważna będzie też odległość jaką pokona sportowiec zasięg kończyn, czy pozycja wykorzystywanych przedmiotów[3]. Inne podejście należy

zastosować jeśli będzie trzeba porównać ruch przedmiotów, gdyż w sporej części przypadków ważna będzie raczej pozycja znaczników względem siebie lub w przestrzeni, a mniej ważny będzie kąt między nimi. Dlatego najpierw trzeba się zastanowić, co będzie porównywane, a następnie trzeba znać cel porównywania ruchów, aby odpowiednio dobrać do niego metodę rejestracji ruchu i jego porównywania.



Rys. 1. Prognoza rynku Motion Capture [4]

Według opisu raportu, w 2023 roku rynek systemów Motion Capture wyniesie około 187 miliardów dolarów, a współczynnik wzrostu (CAGR) między 2017 a 2023 około 9% (Rys. 1).

3. Aplikacja i narzędzia

Aplikacja wykorzystana w badaniu została napisana w języku C# z wykorzystaniem biblioteki C3D.NET oraz technologii WPF (Windows Platform Forms). Program ma za zadanie obliczać współczynniki korelacji Pearsona oraz błędu średnio-kwadratowego dla wczytanego pliku C3D.

C3D.NET jest darmową biblioteką zawierającą klasy i metody do obróbki plików C3D. Jest rozpowszechniana na zasadzie licencji MIT [5].

Projekt pliku C3D był pierwotnie napędzany potrzebą wygodnego i wydajnego formatu do przechowywania danych zebranych w środowisku przechwytywania ruchu. Format przechowuje współrzędne 3D i dane liczbowe dla każdej próby pomiarowej, ze wszystkimi różnymi parametrami opisującymi dane, w jednym pliku. Eliminuje to potrzebę składowania danych o ruchu z dodatkowymi notatkami i informacjami testowymi (które zwykle są oddzielane w dalszym procesie) [6].

4. Sposoby porównywania

Istnieje wiele sposobów porównywania do siebie ruchów, w literaturze jednak najczęściej pojawiały się tylko 3 z nich. Najczęściej korzystano z najprostszego, czyli porównywania pozycji każdego punktu. Jest to najprostszy sposób, jednakże jest on mało miarodajny i przydatny tylko w dokładnie określonych warunkach. Za przykład może posłużyć pływanie na ergometrze, gdzie ruch był porównywany w odniesieniu do stabilnego punktu, co umożliwiło wybranie klatek, w których punkt znajdował się najbliżej statycznego markera i wycięcia odpowiednich faz ruchu [7].

Lepszym rozwiązaniem jest zastosowanie algorytmu obliczania błędu średnio-kwadratowego. Jest to sposób bardzo podobny do poprzedniego, jednakże uwydatnia on bardziej różnicę w ruchach. W przeciwieństwie do poprzedniej metody, jako wynik zwracana jest nie tablica a konkretna liczba, dokładniej mówiąc średnia z kwadratu różnicy obu analizowanych ruchów. Daje to możliwość automatycznego wykorzystania wyniku porównania w aplikacji, co rozszerza możliwości jego zastosowania [8]. Wzór prezentuje się następująco:

$$\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \quad (1)$$

gdzie:

- x_i, y_i - i -te wartości obserwacji z populacji X i Y,
- X- pierwszy zbiór danych (ruch wzorcowy),
- Y- drugi zbiór danych (ruch porównywany),
- n - liczba obserwacji (X i Y mają tyle samo obserwacji).

Ostatnim z omawianych sposobów porównywania ruchu będzie współczynnik korelacji liniowej Pearsona. Jest to inaczej współczynnik określający liniową zależność zmiennych losowych, który znalazł dobre zastosowanie w porównywaniu ruchu. Tutaj także wynikiem jest konkretna liczba, dając możliwość wykorzystania jego na przykład w aplikacjach reagujących na ruch człowieka [9]. Wzór na współczynnik korelacji liniowej Pearsona:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (2)$$

gdzie:

- x_i, y_i - i -te wartości obserwacji z populacji X i Y,
- \bar{x}, \bar{y} - średnie z populacji X i Y,
- X- pierwszy zbiór danych (ruch z którym porównujemy),
- Y- drugi zbiór danych (ruch który porównujemy).

5. Implementacja algorytmów obliczających współczynniki

Algorytm obliczający błąd średnio kwadratowego okazał się dużo łatwiejszy do zaimplementowania i krótszy od korelacji liniowej Pearsona, która posiada dość skomplikowany wzór. Czas wykonania obliczeń był prawie trzykrotnie krótszy w przypadku błędu średnio kwadratowego w stosunku do korelacji liniowej Pearsona gdzie wynosił

Przykład 1. Algorytm obliczania współczynnika błędu średnio-kwadratowego

```
for (Int32 i = 0; i <liczbaKlatek; i++)
{
    //Wczytanie klatek
    C3DFrame frame1 = file1.AllFrames[i];
    C3DFrame frame2 = file2.AllFrames[i];
    for (Int32 j = 0; j <liczbaPunktow; j++)
    {
        //Wczytanie punktów w klatce
        C3DPoint3DData Point3D1 = frame1.Point3Ds[j];
        C3DPoint3DData point3D2 = frame2.Point3Ds[j];

        //Obliczanie sumy kwadratu różnicy punktów
        wskaznik += (Point3D1.X - point3D2.X) *
(Point3D1.X - point3D2.X);
        wskaznik += (Point3D1.Y - point3D2.Y) *
(Point3D1.Y - point3D2.Y);
        wskaznik += (Point3D1.Z - point3D2.Z) *
(Point3D1.Z - point3D2.Z);
    }
}
//Wylczenie wskaźnika błędu średnio-kwadratowego
wskaznik = wskaznik / (liczbaPunktow*liczbaKlatek*3);
```

Opis użytych zmiennych w obu algorytmach (przykład 1 i przykład 2):

- file1,file2-porównywane pliki zawierające analogiczne ruchy osoby ćwiczącej na ergometrze,
- frame1,frame2 - pojedyncze klatki porównywanych plików,
- liczbaKlatek - liczba klatek obu plików (w przypadku różnej liczby klatek w plikach jest brana pod uwagę liczba klatek krótszego pliku),
- liczbaPunktow- liczba punktów w pliku.

Przykład 2. Algorytm obliczania współczynnika błędu średnio-kwadratowego korelacji liniowej Pearsona

```
// Obliczenie średniej dla każdego wymiaru w punktach
for (Int32 i = 0; i <liczbaKlatek; i++)
{
    C3DFrame frame1 = file1.AllFrames[i];
    C3DFrame frame2 = file2.AllFrames[i];
    for (Int32 j = 0; j <liczbaPunktow; j++)
    {
        C3DPoint3DData Point3D1 = frame1.Point3Ds[j];
        C3DPoint3DData Point3D2 = frame2.Point3Ds[j];
        //Średnia punktu we wszystkich klatkach
        x1sr[j] += Point3D1.X / liczba;
        y1sr[j] += Point3D1.Y / liczba;
        z1sr[j] += Point3D1.Z / liczba;
        x2sr[j] += Point3D2.X / liczba;
        y2sr[j] += Point3D2.Y / liczba;
        z2sr[j] += Point3D2.Z / liczba;
    }
}
//Obliczenie wskaźnika
for (Int32 i = 0; i <liczbaKlatek; i++)
{
```

```

C3DFrame frame1 = file1.AllFrames[i];
C3DFrame frame2 = file2.AllFrames[i];
for (Int32 j = 0; j < liczbaPunktow; j++)
{
    C3DPoint3DData Point3D1 = frame1.Point3Ds[j];
    C3DPoint3DData Point3D2 = frame2.Point3Ds[j];
//Obliczanie sumy z licznika we wzorze korelacji liniowej
//Pearsona
    suma += (Point3D1.X - x1sr[j]) * (Point3D2.X -
x2sr[j]) + (Point3D1.Y - y1sr[j]) * (Point3D2.Y - y2sr[j]) +
(Point3D1.Z - z1sr[j]) * (Point3D2.Z - z2sr[j]);
//Obliczanie pierwszej sumy z mianownika we wzorze
//korelacji liniowej Pearsona
    suma1kwadrat += Math.Pow((Point3D1.X -
x1sr[j]),2) + Math.Pow((Point3D1.Y - y1sr[j]), 2) +
Math.Pow((Point3D1.Z - z1sr[j]), 2);
//Obliczanie drugiej sumy z mianownika we wzorze korelacji
//liniowej Pearson
    suma2kwadrat += Math.Pow((Point3D2.X -
x2sr[j]),2) + Math.Pow((Point3D2.Y - y2sr[j]), 2) +
Math.Pow((Point3D2.Z - z2sr[j]), 2);
}
}
// Wyliczenie wskaźnika korelacji liniowej Pearsona
wskaznik = suma / Math.Sqrt(suma1kwadrat *
suma2kwadrat);

```

6. Badanie

Badanie wykonano z użyciem jednego pliku, z którego zostały wycięte dwa analogiczne ruchy pływania na ergometrze w celu zaobserwowania zachowania algorytmu dla ruchu o takim samym zakresie. Pliki te zostały zapisane w formacie c3d. Następnie porównano pierwsze 10 powtórzeń ćwiczenia z kolejnymi 10 powtórzeniami, a także pierwszy cykl z wprowadzonymi różnymi zakłóceniami, aby sprawdzić odporność algorytmów na możliwe błędy w rejestracji ruchu. Zakłóceń dokonano przez wprowadzenie w kopii ruchu przesunięcia o 5 i o 100 jednostek.

Wyniki działania programu porównującego zapisano w poniższej tabeli 1 w celu lepszej interpretacji.

- 1) Zakres wskaźnika;
- 2) Liczba klatek w porównywanym ruchu;
- 3) Porównanie 10 pierwszych powtórzeń ćwiczenia wykonanego na ergometrze z 10 kolejnymi;
- 4) Porównanie z tym samym ruchem lecz z wprowadzonymi zakłóceniami:
 - a) zakłócenie 4 klatek rozłożonych równomiernie w ruchu
 - i. zwiększenie wartości punktów o 5;
 - ii. zmniejszenie wartości punktów o 5;
 - iii. naprzemienne zwiększenie i zmniejszenie o 5;
 - iv. zwiększenie o 100;
 - v. zmniejszenie o 100;
 - vi. naprzemienne zwiększenie i zmniejszenie o 100;
 - b) zakłócenie 20 klatek rozłożonych równomiernie w ruchu
 - i. zwiększenie o 5;
 - ii. zmniejszenie o 5;
 - iii. naprzemienne zwiększenie i zmniejszenie o 5;
 - iv. zwiększenie o 100;
 - v. zmniejszenie o 100;
 - vi. naprzemienne zwiększenie i zmniejszenie o 100;

- 5) Czas działania w milisekundach;
- 6) Liczba linii kodu;

Tabela 1. Porównanie algorytmów dla ruchów o klatkach

Metoda porównania algorytmów	Korelacja liniowa Pearsona	Błąd średniokwadratowy	
1	Od -1 do 1	Od 0 do ∞	
2	2982		
3	-0,8282	86783,68	
4	a	i. 0,99999929681	0,0335345407
		ii. 0,99999929683	0,0335345406
		iii. 0,99999929584	0,0335345402
		iv. 0,99971875	13,4138162
		v. 0,99971893	13,4138161
		vi. 0,99971847	13,4138,163
	b	i. 0,99999650291	0,167672,702
		ii. 0,99999650283	0,167672,703
		iii. 0,99999647922	0,167672,701
		iv. 0,9986044	67,069081137
		v. 0,9986037	67,069081171
		vi. 0,9985944	67,069081170
5	77,0696	28,0354	
6	31	14	

7. Wnioski

Algorytm obliczania korelacji liniowej Pearsona można łatwiej wykorzystać w aplikacjach, gdyż jego wyniki można przekonwertować na wartości procentowe. Dodatkowo algorytm wykrywa, czy ruch jest odbiciem lustrzanym przy ujemnych wartościach wskaźnika, co również może znaleźć ciekawe zastosowanie.

Z kolei algorytm wyliczania współczynnika błędu średniokwadratowego pokazuje za to w bardziej widoczny sposób wszelkie wychylenia porównywanego ruchu w stosunku do głównego ruchu. Współczynnik ten jest jednak mało przejrzysty ze względu na brak górnej granicy jego wartości. Jego obliczenie jest za to szybsze o około 49 ms, a sam algorytm prostszy od korelacji liniowej Pearsona, dzięki czemu łatwo go zamieścić w aplikacji.

Literatura

- [1] M. Kitagawa, B. Windsor. MoCap for Artists: Workflow and Techniques for Motion Capture 1st Edition, 2008.
- [2] W. Chwała, P. Maciejasz. Inżynieria biomedyczna Księga współczesnej wiedzy tajemnej w wersji przystępnej i przyjemnej. Uczelniane Wydawnictwa Naukowo-Dydaktyczne AGH, Kraków 2008.
- [3] W. Chwała, P. Maciejasz. Fascynacja ruchem – technika Motion Capture, Podstawy inżynierii biomedycznej. Wydawnictwo AGH, Kraków, 2009.
- [4] 3D Motion Capture market forecast <https://www.marketresearchfuture.com/reports/3d-motion-capture-system-market-3026> [07.10.2019].
- [5] C3D.NET library <https://archive.codeplex.com/?p=c3d> [07.10.2019]
- [6] About C3D file <https://www.c3d.org/> [07.10.2019].
- [7] K. Kowalczyk, M. Skublewska-Paszowska. Metoda docinania faz ruchu podczas wiosłowania na ergometrze na podstawie danych trójwymiarowych. Czasopismo Journal of Computer Sciences Institute, nr 5, s. 155-158, 2017.

- [8] T. Cloete. Benchmarking full-body inertial motion capture for clinical gait analysis. Stellenbosch University, 2009.
- [9] K. Zaborska., K. Jochymczyk-Woźniak, P. Wodarski, A. Bieniek, A. Porównanie systemów przestrzennej analizy ruchu na przykładzie systemów optycznych i akcelerometrycznych. Czasopismo Aktualne Problemy Biomechaniki, nr 9, s 129--134, 2015.