

Tomasz GRUDZIŃSKI, Jakub GRUDZIŃSKI, Adrian DĘBOWSKI
Politechnika Śląska, Instytut Informatyki

METODY ANIMACJI WIRTUALNYCH POSTACI W INTERAKTYWNYM ŚRODOWISKU TRÓJWYMIAROWYM

Streszczenie. W publikacji przedstawiono kierunki badań prowadzonych na świecie w zakresie sterowania ruchem wirtualnej postaci ludzkiej oraz zarysowano możliwe drogi rozwoju tematu. Położono nacisk na zagadnienia związane z wpływaniem na ruch postaci obliczany w czasie rzeczywistym.

Słowa kluczowe: Wirtualna postać, planowanie ruchu, przechwytywanie ruchu, edycja ruchu, odwrotna kinematyka, szkielet, animacja szkieletowa

THE METHODS OF VIRTUAL CHARACTERS ANIMATION IN AN INTERACTIVE 3D ENVIRONMENT

Summary. The paper presents the research directions in the subject of virtual characters animation control. The possibilities of future research are given. The emphasis is put on the issues of interactive animation computed in real time.

Keywords: Virtual characters, motion planning, motion capture, motion editing, inverse kinematics, skeleton, skeletal animation

1. Wprowadzenie

Szybki rozwój grafiki i animacji komputerowej w ostatnich latach zwiększył zapotrzebowanie na skuteczne techniki animacji wirtualnych postaci ludzkich. Najczęściej stosowane techniki bazują na rejestracji ruchu rzeczywistego modelu (aktora), który przy zastosowaniu techniki określanej jako Motion Capture (MoCap) [13, 15] jest digitalizowany i wykorzystywany do sterowania ruchem postaci wirtualnej, tworzeniu animacji ręcznie przy zastosowaniu specjalizowanych pakietów graficznych i wreszcie kombinacji obu tych podejść. Otrzy-

mywane w ten sposób rezultaty, choć wysokiej jakości, wiążą się z wysokimi kosztami uzyskania i są bardzo czasochłonne. Ponadto, ich istotną wadą jest trudność parametryzacji wyjściowego ruchu, przez co dość trudno jest wykorzystać tego typu animacje w środowiskach interaktywnych, gdzie liczy się elastyczność podejścia – na przykład w grach komputerowych.

Animacje generowane w czasie rzeczywistym na podstawie określonych liczbowych danych są rzadkością, ze względu na mało realistyczny wygląd wynikający ze stosowanych technik, w większości opartych na różnych implementacjach algorytmów odwrotnej kinematyki (IK) [5, 12]. Na ogół wynikowe animacje, chociaż spełniają założenia od strony matematycznej, w bardzo ograniczonym stopniu przypominają ruch, jaki wykonałby w danej sytuacji człowiek. Równania odwrotnej kinematyki mają nieskończenie wiele poprawnych rozwiązań i bez dodatkowych mechanizmów bardzo rzadko dają w wyniku oczekiwany efekt. Powoduje to potrzebę poszukiwania rozwiązań, które mogą wesprzeć algorytmy odwrotnej kinematyki czyniąc, generowane animacje bardziej zbliżonymi do rzeczywistości, a przez to do oczekiwań obserwatora.

2. Główne kierunki badań

Chociaż algorytmy odwrotnej kinematyki pozwalają rozwiązać w sensie matematycznym większość problemów, same w sobie nie są metodą wystarczającą. Systemy animacji opracowywane na świecie można ogólnie podzielić na trzy podstawowe rodzaje – modele proceduralne, modele dynamiczne i modele bazujące na przykładach. W ostatnich latach powstaje też coraz więcej modeli hybrydowych, łączących cechy kilku podejść.

2.1. Modele proceduralne

Znakomitą większość zadań animacyjnych można podzielić na mniejsze podzadania odpowiedzialne za rozwiązywanie konkretnych problemów, a składanie wyników wykonania tych podzadań zostawić na później. Pojedyncze podzadania mogą być rozwiązywane przez tworzenie kontrolerów ruchu opartych na algorytmach prostej i odwrotnej kinematyki wspartych np. ograniczeniami ruchu stawów opisanymi m.in. w [1, 5]. Takie kontrolery ruchu oparte na kinematyce są szybkie i dość łatwe do kontroli i parametryzacji, ponieważ nie uwzględniają żadnych fizycznych własności postaci, nie symulują pracy mięśni, a pracują tylko na konkretnych pozach szkieletu.

Problemem tego podejścia jest trudność w tworzeniu kontrolerów, ponieważ są one przeznaczone dla konkretnych rodzajów ruchu, jak marsz, bieg, skok itp. Ponadto, tworzenie

tego typu mechanizmów jest praktycznie niewykonalne dla profesjonalnych grafików – animatorów, ponieważ jest oparte na matematyce i niuansach programistycznych, czyli technologiach bardzo odległych od typowego środowiska pracy grafika. Dodatkowym minusem tego podejścia jest niemożność wplecenia różnych stylów ruchu do pojedynczego kontrolera, tak aby dla różnych postaci generowany ruch wyglądał inaczej.

2.2. Modele dynamiczne

Część naukowców próbuje osiągnąć pożądane efekty w generowaniu ruchu postaci, bazując na możliwie dokładnej symulacji zjawisk fizycznych zachodzących podczas ruchu [16, 17]. Podstawą takiej symulacji jest reprezentacja systemów – w tym wypadku postaci – jako zestawu mniejszych, sztywnych obiektów geometrycznych posiadających pozycję, rotację oraz prędkości liniowe i kątowe. Osoba lub program sterujący ruchem postaci musi w takim wypadku nadać jedynie początkowe wartości tym parametrom, a następnie określić siły zewnętrzne działające na nie. Dalsze przekształcenia systemu (tu: ruch postaci) zachodzą automatycznie przez obliczenia zgodne z fizyką Newtonowską; obiekty poruszają się pod wpływem sił działających na nie.

Taka metoda może być skuteczna, jeśli na obiekty działają wyłącznie siły zewnętrzne lub siły międzyobiektowe, jak np. tzw. sprężynki (ang. *springs*) w systemach symulacji ubrań. Jeżeli jednak – jak w przypadku postaci – niezwykle ważny element stanowią siły wewnętrzne sterujące ruchem szkieletu, czyli mięśnie, sytuacja robi się znacznie bardziej skomplikowana. Algorytmy takie muszą skądś pobierać informacje o tym, jak do modelu postaci zastosować siły generowane przez ruch mięśni, a to sprawia, że zastosowanie ich do sterowania ruchem postaci jest praktycznie niemożliwe. Dodatkowym kłopotem przy modelu dynamicznym jest trudność w określeniu stanu docelowego postaci – w prosty sposób można jedynie określić warunki początkowe, potem wszystko zależy już od wyniku działania symulatora.

Jednym z podejść do tych problemów są tzw. ograniczenia czasoprzestrzenne (ang. *spacetime constraints*), które pozwalają określić, jak ma wyglądać postawa postaci w kluczowych momentach ruchu. System dba wtedy, by ruch między tymi klatkami kluczowymi był zgodny z zasadami fizyki. Chociaż podejście takie może złagodzić nieco skutki problemów opisanych wyżej, nadal nie jest satysfakcjonujące i to z kilku powodów. Symulacja fizyczna jest czasochłonna, co utrudnia użycie jej w aplikacjach czasu rzeczywistego (np. gry komputerowe), ponadto jeśli dane zadanie ma prawidłowe rozwiązania, dodatkowy czas obliczeniowy potrzebny jest na wybranie rozwiązania najbardziej satysfakcjonującego. Na ogół jest to realizowane przez numeryczną funkcję celu, którą system stara się minimalizować równocześnie z obliczaniem kolejnych kroków symulacji. Chociaż technika ta

pozwała na sterowanie ruchem postaci do pewnego stopnia poprzez zmianę funkcji celu, natrafia na wcześniej wspomnianą przeszkodę wysokiej złożoności czasowej obliczeń oraz jest trudna do realizacji w przypadku postaci o złożoności zbliżonej do złożoności organizmu człowieka.

2.3. Modele bazujące na przykładach

Oba wyżej przedstawione podejścia bazują na tworzeniu nowego ruchu bez potrzeby analizy ani znajomości istniejących animacji. Chociaż podejście takie jest atrakcyjne, ponieważ nie wymaga pracy grafika, wyniki przedstawiane w pracach bazujących na tych metodach wskazują, że trudno osiągnąć wysoki poziom realizmu bez wsparcia profesjonalnych animatorów lub animacji uzyskanych za pomocą sprzętu do przechwytywania ruchu (Motion Capture). Coraz więcej prac prowadzonych obecnie opiera się na wykorzystaniu bazy prawidłowych animacji do tworzenia nowych, unikalnych ruchów [5, 8, 13, 14].

Techniki bazujące na przykładach (ang. *example-based*) wykorzystują szeroki zbiór poprawnych animacji jako podstawę do generowania ruchu, po czym wybrane elementy z tej bazy modyfikują tak, by dopasować ją do zewnętrznych uwarunkowań. Wśród różnych podejść do tematu modyfikacji istniejącego ruchu należy wymienić łączenie animacji (ang. *motion blending*), gdzie kilka ruchów bazowych łączonych jest w jeden wynikowy na zasadzie interpolacji między ruchami bazowymi ze zróżnicowanymi (często również zmieniającymi się w czasie) wagami [15], próby usystematyzowania ruchu jako swoistej maszyny stanów (postaw postaci) i grafu przejść między tymi stanami, często z wykorzystaniem ukrytych modeli Markowa jako podstawy matematycznej przekształceń [8] czy też modeli hybrydowych, gdzie wyjście od prawidłowej animacji jest jedynie początkiem, a końcowy efekt uzyskuje się poprzez połączenie wniosków i danych statystycznych obliczonych na podstawie poprawnych ruchów z zaawansowanymi technikami bazującymi na wcześniej opisanych algorytmach odwrotnej kinematyki, przestrzeni poprawnych zgięć stawów czy też punktami równowagi stawu [5].

Niewątpliwie wyniki osiągnięte przy użyciu istniejących animacji są na tyle obiecujące, że uzasadnione staje się twierdzenie, że właśnie te metody są najbardziej przyszłościowe i w ich rozwijaniu leży nadzieja na stworzenie uniwersalnego systemu sterującego w sposób interaktywny wirtualną postacią.

W dalszej części artykułu skupiono się na przedstawieniu najciekawszych, zdaniem autorów, prac prowadzonych na świecie nad animacją postaci ludzkiej oraz przedstawiono możliwe kierunki rozwoju omawianych metod.

3. Wybrane metody generowania ruchu

W ostatnich latach interaktywne generowanie ruchu stało się tematem wielu prac naukowców na całym świecie. Duża część tych prac koncentruje się na metodach, które można zaliczyć do rodziny metod opartych na przykładach lub do nich zbliżonych. Poniżej zaprezentowano przegląd najciekawszych prac, dotyczących tego tematu.

Brand i Hertzmann opisują w [8] bardzo ciekawą metodę generowania animacji zgodnej ze stylem zachowania określonym na bazie poprawnych, przykładowych animacji, wykorzystując czysto statystyczne techniki, bazujące na Ukrytych Modelach Markowa [20]. Autorzy opisują, jak na podstawie czystych danych z systemu Motion Capture stworzyć model bazowy ruchu wykorzystywany jako punkt odniesienia oraz modele ruchu różniące się stylem, tak, aby można było generować unikalne animacje dla postaci, różniące się szczegółami nieistotnymi z punktu widzenia generatora, ale bardzo poprawiającymi końcowy odbiór animacji przez człowieka. Różnice stylistyczne między różnymi zbiorami danych wejściowych wykrywane są z wykorzystaniem techniki analizy statystycznej PCA. Za jej pomocą wykrywane są najważniejsze stylistyczne stopnie swobody. Opisana przez autorów metoda niestety nie jest przeznaczona do pracy w czasie rzeczywistym, a raczej jako metoda wspierająca edycję animacji w programach graficznych.

Bardzo podobną technikę proponują autorzy w [22]. Również w tej pracy proponowane jest użycie analizy statystycznej PCA na rozległym zbiorze poprawnych animacji do wykrycia stylistycznych stopni swobody, a następnie dzięki modyfikacji wag wektorów własnych tworzenia nowych animacji różniących się stylem od ruchów wejściowych. Praca dotyczy tylko animacji przemieszczania się, w związku z czym trudno uznać ją za uniwersalną, jest jednak kolejnym przykładem użycia statystycznej analizy poprawnych danych do generacji nowych, unikalnych animacji.

Technikę PCA stosuje również Johnson w swojej pracy doktorskiej [5]. Praca jest bardzo obszernym podsumowaniem zagadnień związanych z generowaniem animacji. Autor wychodzi od udowodnienia, dlaczego najlepszym narzędziem matematycznym w pracach nad animowaniem wirtualnych postaci są kwaterniony – trójwymiarowe rozszerzenie liczb zespolonych – a dokładnie ich podzbiór – kwaterniony o jednostkowej długości. Udowodniwszy to, opisuje numeryczny algorytm odwrotnej kinematyki CCD omówiony w [2] i wyjaśnia jego kwaternionową wersję. Dzięki zastosowaniu mapowania logarytmicznego kwaternionów i wykładniczej operacji odwrotnej pokazuje, jak z nieliniowej przestrzeni rotacji w S^3 przejść do liniowej przestrzeni stycznej R^3 , gdzie mają zastosowania zasady liniowej geometrii euklidesowej i można stosować obliczenia statystyczne, zakładające liniową przestrzeń danych wejściowych. Wychodząc z tego autor buduje statystyczny model stawu z punktem równowagi oraz limitami zgięcia określonymi w przestrzeni kwaternionowej i rozszerza

algorytm CCD tak, aby uwzględniał ograniczenia zgięcia. W ostatniej części swojej pracy Johnson wprowadza pojęcie postaw własnych (ang. *Eigenpostures*), obliczanych za pomocą PCA typowych postaw wirtualnej postaci. W tym punkcie jednak praca się kończy, osiągnięte wyniki nie są zadowalające i jak sam autor stwierdza, wymagają dalszych badań. Praca Johnsona jest jednak niezwykle wartościową lekturą i dobrym punktem wyjścia do dalszych badań nie tylko w temacie analizy PCA wykonywanej na ruchach wirtualnej postaci.

Bardzo ciekawa technika opisana została przez Li i jego współpracowników w [21]. Na przykładzie tańczącej postaci autorzy prezentują rozwiązanie, które określają jako teksturę ruchu, zdefiniowaną jako zestaw tekstonów i ich rozkład w czasie i przestrzeni, charakteryzujący naturę przechwyconego (przez Motion Capture) ruchu. Autorzy opisują tekston jako liniowy system dynamiczny (ang. LDS), a rozkład tekstonów jako macierz prawdopodobieństwa przejść między pojedynczymi tekstonami. Same tekstony oraz relacje między nimi są tworzone na podstawie danych z prawidłowego ruchu. Taka dwupoziomowa technika daje spore możliwości generowania nowych animacji – po pierwsze, można modyfikować pojedyncze tekstony za pomocą standardowych algorytmów prostej i odwrotnej kinematyki czy też na innej zasadzie, modyfikując w ten sposób animację na dużym poziomie szczegółowości, po drugie natomiast można, korzystając z macierzy rozkładu tekstonów, generować nową choreografię przy zachowaniu dużego prawdopodobieństwa realistycznego wyglądu wynikowej animacji. Połączenie możliwości zmian na tych dwóch poziomach szczegółowości daje bardzo duże możliwości tworzenia nowych, unikalnych animacji. Słabością tej metody jest trudność w wygenerowaniu ruchu odległego od wszystkich dostępnych wzorców, zatem w bardzo dużym stopniu zasięg metody zależy od zróżnicowania danych bazowych.

W podobnym kierunku zmiernają Lee, Chai i Reitsma w swojej pracy [23]. Zapropozowane rozwiązanie również bazuje na analizie zbioru poprawnych ruchów postaci ludzkiej pozyskanych za pomocą Motion Capture i podobnie jak poprzednie dzieli analizowane dane na dwa poziomy szczegółowości. Na każdym poziomie szczegółowości tworzone są grafy przejść między stanami animacji, a jako matematyczna podstawa wykorzystywane są po raz kolejny ukryte modele Markowa. Modyfikacje ruchu na obu poziomach szczegółowości pozwalają, tak jak w poprzednio omawianej metodzie, zmieniać animację zarówno na poziomie małych szczegółów, jak i całej choreografii ruchu. Autorzy przedstawiają również trzy interfejsy sterowania animacją dostępne dla użytkownika systemu. Podejście ma niestety te same wady co wcześniej wymieniane – przestrzeń generacji nowych animacji ściśle zależy od rozległości poprawnych danych wejściowych.

Bardzo podobną do poprzednich pracę przedstawiają Lee i Elgammal w [24]. Autorzy sugerują dodatkowo, by zamodelować przejścia między stanami animacji za pomocą ukry-

tych modeli Markowa rzędu wyższego niż pierwszy lub nawet zmiennego. W opisanym rozwiązaniu jednak użyte zostały modele Markowa pierwszego rzędu podobnie jak w dwóch poprzednich. Rozwiązanie ma więc podobne wady i zalety.

Nieco inne, ale również bardzo ciekawe podejście do tematu opisano w [18, 19]. Autorzy prezentują oryginalny system sterowania animacją bardziej zbliżony do rozwiązań opartych na zaawansowanych algorytmach odwrotnej kinematyki i dekompozycji skomplikowanych akcji na prostsze, zgodnie z metodologią opisaną w sekcji 2.1. Silnik animacji FreeWill+ zaimplementowany przez autorów prezentuje bardzo interesującą koncepcję inteligentnej dekompozycji skomplikowanych poleceń wysyłanych do wirtualnej postaci na coraz prostsze akcje aż do zejścia na poziom prostych zadań odwrotnej i prostej kinematyki. Rozwiązanie jest również interesujące z programistycznego punktu widzenia, bowiem system ma budowę komponentową, tzn. składa się z wielu niezależnych komponentów implementujących wspólny interfejs i porozumiewających się ze sobą w celu użycia najlepszych komponentów, umożliwiających wykonanie danej akcji. Dzięki takiej architekturze system jest łatwo rozszerzalny i bardzo elastyczny oraz, co ważne, jest bardzo wydajny – może pracować w czasie rzeczywistym. Chociaż autorzy nie wykorzystują w swoim rozwiązaniu poprawnych animacji jako bazy do generowania nowych ruchów, można sobie wyobrazić rozwinięcie proponowanej architektury właśnie o taką analizę. Wydaje się, że takie rozwiązanie mogłoby połączyć zalety obu podejść.

4. Możliwe problemy i kierunki rozwoju

Żadna z opisanych wyżej lub opisanych w innych źródłach metod nie aspiruje niestety do miana uniwersalnej. Część nie jest przeznaczona do pracy w czasie rzeczywistym, część jest ograniczona w dużym stopniu rozległością danych wejściowych, niektóre można stosować tylko w pewnym ograniczonym spektrum generowanych ruchów. Temat interaktywnej generacji animacji wirtualnej postaci ludzkiej pozostaje zatem wciąż otwarty.

Na podstawie wcześniejszych rozdziałów można stwierdzić z dużą dozą pewności, że nie ma możliwości zrobienia systemu generującego ruch bez wyjścia od znanego zbioru poprawnych ruchów. Analiza bazy danych takich animacji wykonana w celu znalezienia typowych postaw, stylistycznych stopni swobody podobnych animacji czy też zbudowania grafów przejść między kluczowymi ruchami mogłaby być wykonana jednorazowo na etapie przygotowania danych dla silnika animacji pracującego w czasie rzeczywistym. Przy odpowiednim zbiorze danych wejściowych można też, jak dowodzi Johnson w [5], wydobyć informacje o punktach równowagi stawów oraz przestrzeniach poprawnego zgięcia stawów, a co za tym idzie również liczby stopni swobody stawu (1, 2 lub 3).

Takie dane mogłyby być bardzo pomocne jako pomocnicze dane używane później przez silnik animacji pracujący w czasie rzeczywistym. Tutaj bardzo atrakcyjnym pomysłem wydaje się hierarchiczny, wieloagentowy model sterowania animacją przedstawiony w [18]. Dobre rezultaty mogłyby przynieść zastosowanie podobnego, otwartego architektonicznie podejścia programistycznego z jednoczesnym wykorzystaniem omawianych wcześniej przygotowanych już danych do rozwiązywania konkretnych podzadań. Można sobie wyobrazić, że ruch generowany na podstawie działania algorytmu odwrotnej kinematyki wyglądałby znacznie bardziej realistycznie, gdyby jako punkt wyjścia potraktować znaną poprawną postawę zbliżoną do docelowej, a w samym ruchu wykorzystać wiedzę o korelacjach między ruchami sąsiadujących kości czy znajomość limitów zgięć stawów.

Podsumowując, proponowana linia rozwiązania powinna maksymalnie wykorzystywać dane pochodzące z poprawnych animacji, lecz ze względów wydajnościowych ich analiza nie powinna być przeprowadzana na bieżąco, a jedynie na etapie przygotowania danych dla systemu. Tam, gdzie to możliwe, powinny być wykorzystane znane ruchy lub wariacje ich segmentów jak opisano m.in. w [21] z ewentualną parametryzacją za pomocą stylistycznych stopni swobody wykrywanych wcześniej, np. za pomocą analizy PCA. Do niektórych zadań musi być wykorzystywana odwrotna kinematyka, ale w miarę możliwości należy wspierać ją podpowiedziami bazującymi na znanych poprawnych ruchach oraz informacjami o budowie szkieletu, przede wszystkim dodatkowymi parametrami stawów, takimi jak limity zgięć czy też stopień sztywności. Otwarta architektura programistyczna powinna pozwolić na domyślne użycie uniwersalnych algorytmów przy jednoczesnym zachowaniu możliwości automatycznego – w ramach dostępności – użycia wyspecjalizowanych kontrolerów ruchu rozwiązujących konkretne zadania o małym lub średnim stopniu złożoności. Rozwiązanie takie – chociaż niełatwe do zrealizowania – mogłoby dać bardzo dobre efekty i być może nawet aspirować do miana rozwiązania uniwersalnego.

5. Wnioski końcowe

Animacja wirtualnej postaci ludzkiej jest zagadnieniem bardzo złożonym. Jak dotąd nie przedstawiono jeszcze rozwiązania, którego zastosowanie eliminowałoby wszystkie powstające problemy. Niezwykle trudno jest połączyć realizm wykonywanych ruchów z jego interaktywnym parametryzowaniem wykonywanym w czasie rzeczywistym, bez możliwości dostosowania parametrów metodą prób i błędów.

Bez wątplenia podstawowym wnioskiem wynikającym z przedstawionych wyników prac w temacie na całym świecie jest fakt, że nie da się osiągnąć zadowalających wyników, wykorzystując jednostronne podejście do tematu – czy to oparte czysto na dynamice postaci,

czy też na modelu kinematycznym lub wykorzystaniu zbioru poprawnych ruchów. Rozwiązanie aspirujące do miana uniwersalnego musi łączyć zalety rozwiązań cząstkowych minimalizując ich wady.

Niewątpliwie wygenerowanie realistycznie wyglądającego ruchu bez użycia wcześniej przygotowanych danych jest obecnie niemożliwe, zatem z pewnością muszą zostać użyte techniki oparte na przykładach. Wydaje się, że statystyczna analiza ruchu może umożliwić uzyskanie danych, pozwalających na płynną zmianę podstawowych parametrów ruchu, jak chociażby prędkość poruszania się, bez przygotowywania wielu różniących się nieznacznie animacji. Co najmniej przy części zadań niezbędne jest wykorzystanie algorytmów, bazujących na odwrotnej kinematyce, niemniej musi być ona wsparta przez system podpowiedzi bazujący na znajomości typowych ruchów postaci, inaczej bowiem trudno będzie zlikwidować wrażenie „mechaniczności” ruchu.

Konkludując, należy się spodziewać, że badania w temacie animowania interaktywnych postaci iść będą w stronę budowy modeli wykorzystujących bazę danych poprawnych ruchów jak podstawę, ale z możliwością dodatkowego parametryzowania oraz zaburzania tychże dla uzyskania unikalnych animacji.

LITERATURA

1. Herda L., Urtasun R., Fua P., Hanson A.: Automatic Determination of Shoulder Joint Limits using Quaternion Field Boundaries. *International Journal of Robotics Research*, 22(6), June 2003.
2. Mysiek T., Grudziński T., Ross J.: Zaawansowane techniki animacji szkieletowej modeli trójwymiarowych, *Studia Informatica*, 2006.
3. Hanson A.: Constrained Optimal Framings of Curves and Surfaces using Quaternion Gauss Maps. *Visualization*, s. 375÷382. IEEE Computer Society Press, 1998.
4. Perlin K.: Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics*, 1(1), 1995.
5. Johnson M. P.: Exploiting Quaternions to Support Expressive Interactive Character Motion. PhD thesis, Massachusetts Institute of Technology, 2003.
6. Blumberg B. M.: New Dogs, Old Tricks: Ethology and Interactive Characters. PhD thesis, The Media Lab, Massachusetts Institute of Technology, 1997.
7. Grassia F. S.: Believable Automatically Synthesized Motion by Knowledge - Enhanced Motion Transformation. PhD thesis, Carnegie Mellon University, 2000.
8. Brand M., Hertzmann A.: Style machines. *Computer Graphics (Proceedings of SIGGRAPH 2000)*, 2000.

9. Grassia F. S.: Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools*, 3(3) , 1998, s. 29÷48.
10. Lee J.: A Hierarchical Approach to Motion Analysis and Synthesis for Articulated Figures. PhD thesis, Korea Advanced Institute of Science and Technology, 2000.
11. Kuipers J. B.: *Quaternions and Rotation Sequences*. Princeton University Press, 1999.
12. Blow J.: Inverse kinematics with joint limits. *Game Developer Magazine*, 2002.
13. Barbic J., Safonova A., Pan J-Y., Faloutsos C., Hodgins J. K., Pollard N. S.: Segmenting Motion Capture Data Into Distinct Behaviours. In *Proceedings of Graphics Interface 2004 (GI'04)*, London, Ontario, Canada, Maj 17-19, 2004.
14. Kalisiak M., van de Panne M.: A Grasp-based Motion Planning Algorithm for Character Animation. Department of Computer Science, University of Toronto, 2001.
15. Kovar L., Gleicher M.: Flexible Automatic Motion Blending with Registration Curves, *ACM SIGGRAPH*, 2003.
16. Cohen M. F.: Interactive Spacetime Control for Animation. *Computer Graphics*, vol. 26, July 1992, s. 293÷302.
17. Baraff D., Witkin A.: Large Steps in Cloth Simulation. *Computer Graphics*, July 1998, s. 43÷54.
18. Francik J., Szarowicz A.: Character Animation with Decoupled Behaviour and Smart Objects. 6th International Conference on Computer Games CGAIMS, 27 – 30 July, Louisville, Kentucky, USA 2005, Extended Paper.
19. Francik J., Szarowicz A.: Integrate and Conquer – the Next Generation of Intelligent Avatars, *ACM SIGCHI International Conference on Advances in Computer Entertainment Technology ACE 2005*, Special Session on Computer Games Technology 15th - 17th June, Valencia, Spain 2005.
20. Eddy S. R.: What is a Hidden Markov Model? *Nature Biotechnology* 22, 2004, s. 1315÷1316.
21. Li Y., Wang T., Shum H.-Y.: Motion Texture: A Two-Level Statistical Model for Character Motion Synthesis. *Proceedings of ACM SIGGRAPH*, 2002.
22. Urtasun R., Fua P.: Mahalanobis Motion Generation.
23. Lee J., Chai J., Reitsma P.: Interactive Control of Avatars Animated with Human Motion Data. *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 2002.
24. Lee C.-S., Elgammal A.: Human Motion Synthesis by Motion Manifold Learning and Motion Primitive Segmentation. *Lecture Notes in Computer Science*, 2006.

Recenzent: Dr hab. inż. Maria Pietruszka, prof. Pol. Łódzkiej

Wpłynęło do Redakcji 27 listopada 2007 r.

Abstract

Animating virtual human-like characters has been a challenge in computer graphics for many years. Until now no single technique exists which addresses all of the arising problems. This paper presents a short summary of the methods and directions that were taken around the world to develop a solution to the subject. The most popular example-based algorithms, which make use of motion capture techniques, are covered. Some model-based solutions, operating on kinematical, dynamical or hybrid mathematical human models are introduced as well. The most important low-level subsystems like inverse kinematics solvers or motion blending algorithms are presented.

The paper is meant to be an introduction to the amazing subject of animating virtual human characters and beside summarizing the current state-of-the-art it suggests some future research directions that could be taken to investigate the area even deeper. The conclusions are made to point out which of the methods are most promising and worth further improving.

Adresy

Tomasz GRUDZIŃSKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-100 Gliwice, Polska,

Jakub GRUDZIŃSKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-100 Gliwice, Polska,

Adrian DĘBOWSKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-100 Gliwice, Polska.