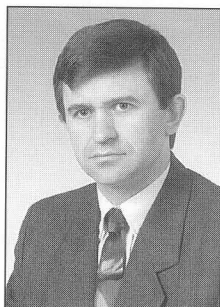


Ryszard LENIOWSKI

POLITECHNIKA RZESZOWSKA, KATEDRA INFORMATYKI I AUTOMATYKI

Zastosowanie grafiki komputerowej do wirtualnego prototypowania robotów

dr Ryszard LENIOWSKI



Absolwent Politechniki Rzeszowskiej. Stopień doktora uzyskał na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej. Wieloletni pracownik naukowo-dydaktyczny Politechniki Rzeszowskiej. Zajmuje się robotyką (konstruktor czterech robotów badawczych i dydaktycznych wraz z systemami sterowania), grafiką komputerową i wirtualnym prototypowaniem. Autor ponad 40 artykułów w języku polskim i 20 w języku angielskim. Twórca kilku systemów automatyki przemysłowej wdrożonych w zakładach przemysłowych i firmach woj. podkarpackiego.

Streszczenie

Praca prezentuje zastosowanie zbioru komponentów programowych do symulacji, animacji i wirtualnego prototypowania operacji technologicznych z użyciem robotów przemysłowych. Przy tworzeniu komponentów wykorzystano technikę programowania obiektowego oraz technologię COM [1]. Animację 3D oparto na standardzie OpenGL [2]. Współbieżnie działające procesy zbudowane z komponentów tworzą środowisko symulacyjne. Wymiana danych pomiędzy procesami realizowana jest przy pomocy interfejsu Winsock. Animacja dokonywana jest na bazie wyników uzyskiwanych z procesu symulacji (modele ODE) oraz geometrycznych modeli urządzeń i zjawisk występujących w wirtualnym gnieździe produkcyjnym. Oprogramowanie ma otwartą architekturę i pozwala dołączać nowe komponenty jeżeli tylko spełniają założenia technologii COM. Jako przykład zastosowania zbioru komponentów zaprezentowano symulację procesu lakierowania metalowej powierzchni z uwzględnieniem szczegółów graficznego modelu robota i jego otoczenia.

Abstract

This paper presents the application of software components collection for simulations, animations and virtual prototyping of technological processes with the use of industrial robots. Object programming technique and COM standard have been used for components design. Three dimensional animation has been supported by OpenGL library. The simulation environment contains concurrent working processes based on the components designed. Data exchange between such working processes is realized on the basis of Winsock interface. The animation uses geometrical machine models and simulations process data. The open architecture software allows to attach any new components if they are fulfill the requirements of COM standard. As an example of the components set application, the simulation of the metal surface varnish process has been presented. This simulation takes into account a great number of graphics details of robot model and its environment.

Słowa kluczowe: symulacja komputerowa, wirtualne prototypowanie, robotyka, grafika komputerowa, sterowanie ruchem.

Keywords: computer simulation, virtual prototyping, robotics, computer graphics, motion control.

1. Wstęp

Projektowanie, uruchamianie i obsługa nowoczesnych systemów produkcji i wytwarzania, opartych o zrobotyzowane linie technologiczne oraz skomplikowane układy automatyki przemysłowej, stawia wysokie wymagania przed kadrą inżynierską. Grupa ta powinna zdobyć podczas studiów niezbędną wiedzę teoretyczną, popartą doświadczeniem laboratoryjnym. Realizację ambitnych planów edukacyjnych utrudnia jednak brak dostępu do laboratoriów wyposażonych w zrobotyzowane gniazdo produkcyjne. Nawet w zamożnych krajach Europy, Japonii i USA laboratoria uniwersyteckie dysponują zaledwie pojedynczymi egzemplarzami robotów i skromnym otoczeniem technicznym. Wydaje się, że jedynym sposobem zapełnienia luki pomiędzy stanem laboratoriów uniwersyteckich i nowoczesnym przemysłem jest stosowanie wirtualnych środowisk symulacyjnych. Współcześnie, komputerowo wspoma-

gane prototypowanie jest jednym z etapów nowoczesnego badania urządzeń, maszyn i systemów zrobotyzowanych. Próbuje się połączyć metody projektowania oferowane przez pakiety CAD z możliwościami symulacyjnymi programów jakie posiada np. MATLAB. Symulacje ruchu maszyn z uwzględnieniem operacji technologicznych generują znaczne ilości danych, których obserwacja za pomocą tradycyjnych interfejsów użytkownika jest uciążliwa. Dlatego też, istotnym elementem środowiska programowego jest interfejs graficzny z modułem animacji typu 3D. Obecnie, dąży się do obserwacji pracy wirtualnego maszyny w jego typowym otoczeniu. Spełnienie wymienionych oczekiwań jest możliwe dzięki nowoczesnym bibliotekom graficznym oraz specjalizowanym procesorom GPU. Istniejące komercyjne oprogramowanie symulacyjne nie nadąża jednak za stale wzrastającymi wymaganiami i potrzebami. Powodem jest złożoność analizowanych procesów, ich dynamiczny charakter, zmienność oddziaływań zewnętrznych i coraz bardziej wyrafinowana interpretacja graficzna. Autor proponuje więc rozwiązanie, polegające na opracowaniu otwartej biblioteki komponentów programowych przeznaczonych do symulacji, animacji i wirtualnego prototypowania operacji technologicznych z użyciem robotów przemysłowych. Przy tworzeniu komponentów wykorzystana została technika programowania obiektowego oraz technologia COM. Animację 3D oparto na standardzie OpenGL, a współbieżnie działające procesy, zbudowane z komponentów, tworzą środowisko symulacyjne. Wymiana danych pomiędzy procesami realizowana jest przy pomocy interfejsu Winsock. Animacja wykorzystuje geometryczne modele urządzeń i zjawisk występujące w wirtualnym gnieździe produkcyjnym i realizowana jest na bazie wyników pochodzących z procesu symulacji (modele ODE) [3]. Oprogramowanie ma otwartą architekturę i pozwala dołączać nowe komponenty, jeżeli tylko spełniają założenia technologii COM. Jednym z ważniejszych elementów jakie zawiera przygotowane oprogramowanie jest moduł wykrywania kolizji, które towarzyszą wszystkim operacjom technologicznym. Wykorzystując opracowane komponenty stworzono oryginalną interpretację graficzną (modele) kilku procesów technologicznych.

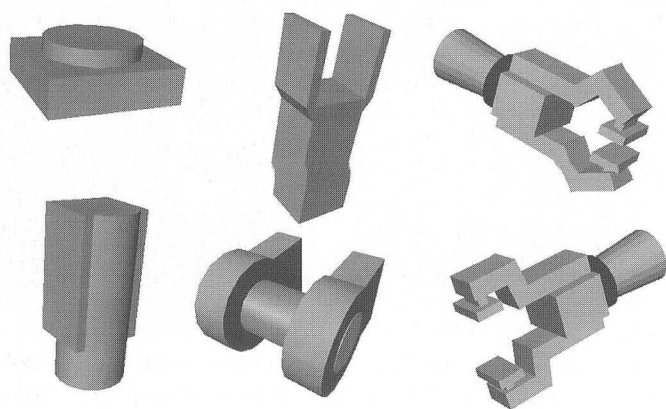
2. Robot przemysłowy jako komponent programowy

Problem wizualizacji poruszających się manipulatorów jest ściśle związany z kinematyką układu, czyli wszystkimi geometrycznymi i czasowymi własnościami ruchu, bez wnikania w przyczyny jego powstania [4]. Model kinematyki robota stanowi podstawę budowy jego reprezentacji geometrycznej.

Biblioteka komponentów została zaprojektowana w sposób umożliwiający tworzenie modeli robotów różnych typów (zgodnie z zasadą maksymalnej uniwersalności) przy wykorzystaniu OOP (ang. Object Oriented Programming). W tym celu wyodrębniono abstrakcyjną klasę: cRobot, reprezentującą uogólnione pojęcie robota przemysłowego. Na jej bazie utworzono klasy cKrepy, cFlex3D, cMikrus reprezentujące roboty badawcze lub dydaktyczne skonstruowane przez autora, cKrat (robot kratownicowy) lub roboty przemysłowe wybranych producentów (klasy cIRp6, cIRb2400,...).

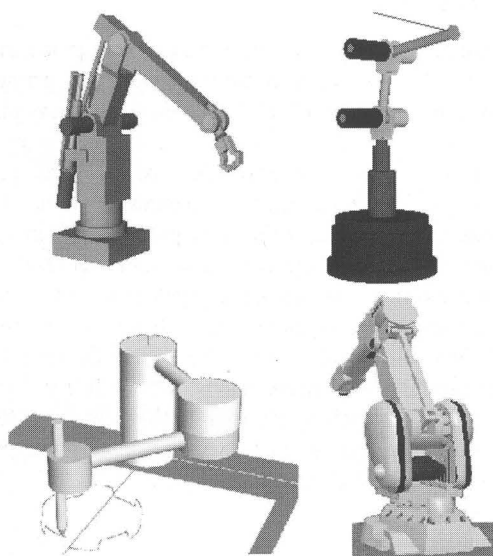
Klasa cRobot specyfikuje interfejs wspólny dla wszystkich robotów przemysłowych, zaś klasy reprezentujące konkretne modele zawierają właściwą implementację tego interfejsu. Na przykład wspólną cechą szerokiej klasy manipulatorów jest możliwość ste-

rowania przegubem. Stąd w klasie abstrakcyjnej znajduje się metoda pozwalająca na dokonanie tej czynności. Ponieważ w różnych modelach manipulatorów ruch przegubem jest związany z jego strukturą kinematyczną, dlatego implementacja odpowiednich metod znajduje się w klasach reprezentujących konkretny typ robota. Klasa abstrakcyjna *cRobot* posiada zbiór 10 metod tworzący publiczny interfejs tej klasy i wszystkich klas z niej dziedziczących (np. *readPos* - odczytanie stanu przegubu). Dzięki temu sterowanie każdym rodzajem robota od strony aplikacji przebiega identycznie. Oprócz publicznego interfejsu klasa *cRobot* zawiera metodę chronioną *makeGrafModel* odpowiedzialną za graficzną prezentację urządzenia opracowaną przy wykorzystaniu biblioteki OpenGL. Pomimo znacznej różnorodności konstrukcji układu kinematycznego robotów przemysłowych można zebrać wiele wspólnych podzespółów, których rzeczywisty kształt przybliży kilka rodzin brył geometrycznych modelujących korpusy, ramiona, napędy, przeguby, chwytaki czy prowadnice (rys.1).



Rys.1. Elementy składowe robotów.
Fig.1. Robots components.

Opracowany zestaw elementów konstrukcyjnych upraszcza opracowywanie wirtualnych modeli robotów badawczych, przemysłowych lub dydaktycznych (rys.2).



Rys.2. Roboty utworzone z elementów składowych: iRp6, Krępy, Mikrus, iRb2000.

Fig.2. Robots iRp6, Krępy, Mikrus, iRb2000 assembled with elementary components.

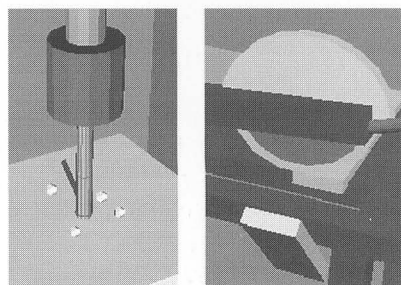
Aby zwiększyć realizm modelu graficznego wprowadzono oryginalną kolorystykę i teksturowanie odpowiadające polerowanym powierzchniom metalowym i detalom wykonanym z tworzyw sztucznych. Cieniowanie powierzchni zrealizowano w oparciu o model Phonga.

3. Animacja procesu technologicznego

Roboty są jednymi z wielu elementów składowych, które wraz z innymi urządzeniami pomocniczymi, tworzą nowoczesny ciąg

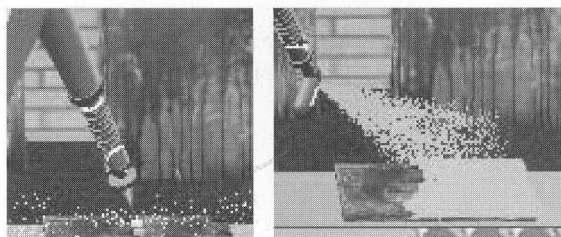
technologiczny. Należą do nich: maszyny (taśmociągi, podajniki, ...), wyposażenie (ograniczniki, prowadnice, szafy sterownicze), narzędzia (spawalnice, pistolety, zgrzewarki, ...), opakowania (pojemniki, palety, zbiorniki, beczki, butle), osłony (kurtyny bezpieczeństwa, bariery) i wiele innych. Oprócz elementów stale obecnych, na scenie mogą pojawić się obiekty, których czas przebywania nie jest zdeterminowany. Nierzadko scena składa się z kilkuset obiektów, o łącznej liczbie wielokątów składowych przekraczającej 10^6 . Dlatego też bardzo ważną czynnością jest optymalizacja kodu oraz sposobu wyświetlania tylko tych obiektów, które są aktualnie widoczne przez obserwatora, co znacząco zwiększa wydajność animacji. W prezentowanej bibliotece komponentów proponowane są dwie metody. Pierwsza polega na podziale przestrzeni na obszary z przypisanym do nich zestawem kamer, druga zaś wykorzystuje algorytm portali. Polega on na podziale sceny na sektory, które zawierają niewidoczne dla obserwatora dodatkowe trójkąty zamykające, zwane portalami.

Spośród wielu różnych procesów technologicznych można wyróżnić takie, których model graficzny można opracować względnie łatwo, oraz takie, których modelowanie jest bardzo skomplikowane. Do pierwszej grupy możemy zaliczyć np. modele emaliowania lub malowania przez zanurzenie. Przedmiot (model bryłowy) uzyskuje wtedy kolor i atrybuty powierzchni, które przypisano wirtualnej farbie. Trudniej modeluje się procesy montażu, obróbki powierzchniowej takiej jak wiercenie, skrawanie, cięcie lub zgrzewanie. Należy wtedy wytworzyć wirtualne wióry lub opiłki, których stan powinien odpowiadać rzeczywistości oraz przewidzieć co stanie się z przeciętym detalem (np. upadek do pojemnika) lub spoiną (zmiana koloru na skutek stygnięcia detalu). Modele graficzne wiercenia i cięcia pokazuje rys.3.



Rys.3. Modele graficzne wiercenia i cięcia detalu.
Fig.3. Graphical models of drilling and cutting process.

Znacznie trudniej tworzy się modele graficzne procesów technologicznych, w wyniku których powstaje wiele efektów ubocznych (światło, dym, pył). Przykładowo, należą do nich spawanie łukowe czy malowanie pistoletowe. Spawanie łukowe wydziela znaczną ilość światła o zmiennym, losowym natężeniu, która musi zostać uwzględniona w globalnym modelu oświetlenia. Przy skierowaniu kamery na spawalnicę powstaje zjawisko oślepienia obserwatora. Dodatkowym efektem jest unoszący się dym, oraz snopy iskier stygnących w powietrzu. Proponowany model spawania, który jest dostępny w bibliotece komponentów umożliwia definiowanie współrzędnych (miejsca) operacji, ustalenie koloru i intensywności dymu, wybrania rodzaju i natężenia iskier oraz ustawienia flagi „lens flare“ modelującej efekt oślepienia. Przykładowy proces spawania pokazano na rys.4.



Rys.4. Modele graficzne spawania i lakierowania.
Fig.4. Graphical models of welding and varnishing process.

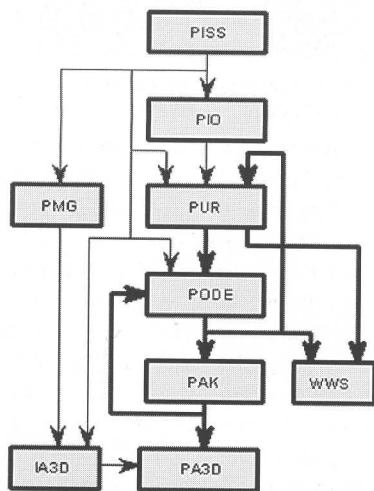
W modelu graficznym malowania natryskowego zaproponowano stożek rozpylania farby z kątem odchylenia cząstek aerozolu od osi dyszy pistoletu wybieranym losowo. Aktywna cząstka porusza się wtedy w polu grawitacyjnym do momentu napotkania przeszkody (powierzchni). W momencie wykrycia kontaktu z powierzchnią identyfikowany jest segment siatki, który przyjmuje atrybuty farby. Uzyskany efekt pokazano na rys.4.

4. Modelowanie kolizji w przestrzeni roboczej

Do opisu wielu operacji technologicznych można wykorzystać zjawisko kolizji. Kolizja robota z obiektem statycznym polega na zetknięciu poruszającego się robota z wyposażeniem linii lub gniazda produkcyjnego. Dla celów detekcji kolizji skomplikowane modele geometryczne sprowadza się najczęściej do prostych brył, których ścianami są trójkąty i czworokąty. Detekcja kolizji polega wtedy na sprawdzeniu czy figury definiujące obiekty kolizyjne posiadają część wspólną. Aby rozwiązać ten problem stosowane są różne własności znane z geometrii analitycznej. W opracowanym oprogramowaniu wykorzystano własność badania wzajemnego położenia 3 punktów za pomocą wyznacznika macierzy. Zaletą algorytmu jest prostota. Problem pojawia się w przypadku istnienia dużej liczby obiektów, z którymi robot może wejść w kolizję. Problem ten rozwiązano wprowadzając podział przestrzeni na sektory, tak aby w każdym znajdowała się niewielka liczba obiektów bryłowych wchodzących w kolizję. Stosuje się model sąsiedztwa ośmiokierunkowego, który generuje dziewięć sektorów kolizyjnych. W przypadku kolizji z obiektami dynamicznymi bada się dodatkowo odległość pomiędzy współrzędnymi ich otoczki i uproszczoną bryłą robota. Obliczenia wykonuje się jedynie dla tych obiektów, które znajdują się w przestrzeni roboczej manipulatora. Proponowana metoda zapobiega niekontrolowanemu wniesieniu części robota w powierzchnię obiektu.

5. Architektura oprogramowania

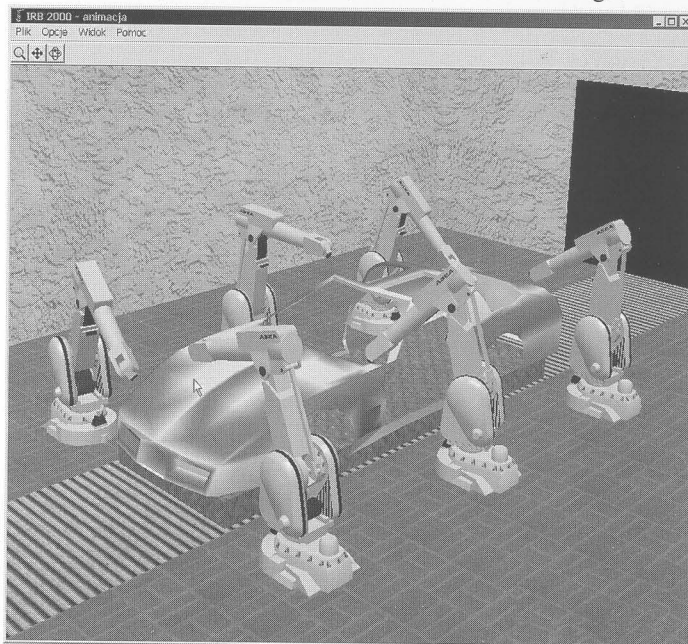
Zbudowane przy pomocy kolekcji komponentów środowisko symulacyjne podzielono na szereg odrębnych procesów, z których każdy składa się z wątków programowych. Poglądowy schemat struktury środowiska symulacyjnego pokazuje rys.5.



Rys.5. Architektura oprogramowania
Fig.5. Architecture of the software.

W powyższym schemacie można wyodrębnić siedem procesów, które wymieniają pomiędzy sobą dane za pomocą interfejsu Winsok. Głównym źródłem danych jest proces PODE (zawiera modele matematyczne robotów, urządzeń i zjawisk przeznaczonych do symulacji a także procedury numeryczne rozwiązujące modele dynamiki) oraz proces układu regulacji. Z danych korzystają procesy analizy kolizji-PAK, wizualizacji wektorów stanu urządzeń- WWS oraz animacji 3D- PA3D. Sterowanie przebiegiem symulacji odbywa się z poziomu procesu operatora- PIO, który odpowiada za dialog użytkownika ze środowiskiem symulacyjnym. Poprzez interfejs animacji-IA3D moż-

liwe jest wybranie trybu i scenariusza prezentacji, ustawienia parametrów oświetlenia oraz aktywowanie flag wizualizacji. Proces wykrywania kolizji wykorzystuje uproszczone modele bryłowe urządzeń. Uzupełnieniem środowiska symulacyjnego jest oprogramowanie komunikacyjne oraz komponenty pomocnicze. Na rys.6 pokazano animację procesu lakierowania karoserii samochodu osobowego.



Rys.6. Animacja procesu malowania maski samochodu.
Fig.6. Animation of a car body painting process.

Działanie oprogramowania poprzedza etap konfiguracji i definiowania początkowych wartości parametrów niezbędnych do dalszych obliczeń. Należy dodać, że indywidualne skonfigurowanie komponentów wymaga użycia środowiska programowego takiego jak MS Developer Studio.

6. Podsumowanie - kierunki rozbudowy oprogramowania

Rozległość problematyki związanej z symulacją procesów technologicznych z wykorzystaniem robotów powoduje, że przygotowanie wiarygodnych modeli jest złożone i czasochłonne. Szczególnie trudne do modelowania są takie procesy, w których należy uwzględnić elastyczność lub plastyczność materiałów, bądź opierać model graficzny na mało dokładnym modelu matematycznym. Do równie skomplikowanych należy modelowanie pracy urządzeń mobilnych w zmiennym środowisku. Pojawiło się również zapotrzebowanie na modelowanie graficzne sytuacji awaryjnych oraz ekstremalnych, takich jak pożar lub eksplozja, które mogą zaistnieć w fabryce. Dlatego autor zamierza wprowadzić do zbioru komponentów modele graficzne rozprzestrzeniania się dymu, ognia oraz graficzną interpretację uszkodzeń mechanicznych opartą na modelu fizycznym. Testowane będą także nowe wersje modułu matematycznego zdolne do rozwiązania złożonych równań dynamiki w trybie real-time.

7. Literatura

- [1] S. Li, P. Economopoulos: Professional COM Applications with ATL. Wrox 1998.
- [2] M. Sagal, K. Akeley: The OpenGL Graphics System. Specification V.2.0. 2004.
- [3] J. Popovic, S.M.Seitz, M. Erdmann Z. Popovic, A. Witkin: Interactive Manipulation of Rigid Body Simulations. Proc. of SIGGRAPH 2000, pp. 209-217.
- [4] R. Leniowski, S.Samolej, A. Szumiało, P. Wolański: Animacja robotów przemysłowych przy pomocy biblioteki OpenGL, Prace naukowe ICT, Wrocław 1998, str. 125-132.

Title: Application of Computer Graphics for Robots Virtual Prototyping

Artykuł recenzowany