

# Metody wirtualnej rzeczywistości w modelowaniu układów biomolekularnych i nanoukładów.

Kurs dydaktyczny Uniwersytetu Warszawskiego  
dla studiów II stopnia nauk przyrodniczych  
(wykłady i ćwiczenia)

## *Spis treści:*

<i>I. Bogdan Lesyng, Rozwoj metod wizualizacji 3D i technologii wirtualnej rzeczywistości oraz wyposażenie pracowni.</i>	2
<i>II. Krzysztof Nowiński, Modułowe rozwiązania grafiki komputerowej 2D i 3D oraz wizualnej analizy danych.</i>	5
<i>III. Paweł Daniluk, Programowanie urządzeń haptycznych.</i>	9
<i>IV. Aleksander Dębiński, Zastosowania manipulatorów w modelowaniu prostych układów biomolekularnych i nanoukładów w środowisku NAMD/VMD.</i>	22
<i>V. Krystiana Krzyśko, Wizualizacja 3D wyników symulacji metodami dynamiki molekularnej złożonych struktur biomolekularnych.</i>	28
<i>VI. Magdalena Gruzziel, Metody modelowania złożonych układów biomolekularnych z wykorzystaniem modeli mezoskopowych, m.in. UNRES.</i>	39
<i>VII. Bogdan Lesyng, Mathematica i techniki VR.</i>	43

***Koordinacja prac:*** Bogdan Lesyng

***Podziękowania:*** Powstanie niniejszego kursu było możliwe dzięki wsparciu finansowemu Funduszu Inicjatyw Dydaktycznych Uniwersytetu Warszawskiego, FID 500/11-01-02/04-92.

Materiały w wersji elektronicznej dostępne są na serwerze Centrum Doskonałości UW BioExploratorium <http://kms.bioexploratorium.pl/>.

## **I. Rozwój metod wizualizacji 3D i technologii wirtualnej rzeczywistości oraz wyposażenie pracowni.**

### **Wstęp.**

Postęp w badaniach układów złożonych takich jak układy biomolekularne i nanoukłady nie byłby możliwy bez wykorzystania zaawansowanych metod wizualizacji struktur i pól molekularnych. Pionierskie rozwiązania technik wizualizacji 3D zapoczątkowany był przez krystalografów w latach sześćdziesiątych, tworząc z wykorzystaniem ploterów rzuty trójwymiarowych struktur na płaszczyznę dla lewego i dla prawego oka. Przy pomocy odpowiednio skonstruowanych okularów skracających ogniskową można było oglądać struktury trójwymiarowe. Wcześniej, już od połowy 19 wieku takie lub podobne techniki stosowane były w różnego rodzaju stereoptikonach, np. w prezentacjach architektury.

W ciągu ostatniego dwudziestolecia dokonał się następny, jakościowy postęp, a mianowicie rozwój metod wirtualnej rzeczywistości. Oznacza to m.in. możliwość interaktywnej pracy z generowanymi komputerowo, wirtualnymi trójwymiarowymi obiektami podlegającymi zakodowanym prawom fizyki. Rozwój technologii napędzany jest jak to często bywa potrzebami wojskowymi, rozwojem medycyny a także rozrywki – tym przypadku rozwojem kina 3D.

Program tego wykładu oraz towarzyszących mu na pracowni komputerowej ćwiczeń rozpoczyna się od technik dwu i trójwymiarowej wizualizacji danych jak również wizualizacji wyników zaawansowanych symulacji komputerowych różnego rodzaju złożonych obiektów i procesów fizycznych, poprzez programowanie i używanie manipulatorów (urządzeń haptycznych, *haptic devices*) w badaniach prostych nanostruktur i układów biomolekularnych (nanorurki, C60 z zadokowanym jonem, oligopeptydy) aż do badań złożonych układów (bio)molekularnych. Oprócz modeli nanostruktur i obiektów (bio)molekularnych tworzonych z mikroskopowych atomowych elementów omówione też zostaną wybrane modele gruboziarniste, gdzie podstawowymi elementami z których składają się (bio)molekuły nie są atomy ale grupy atomów. W szczególności zostanie zaprezentowany model UNRES, którego podstawowymi elementami składowymi są całe reszty aminokwasowe oraz wiązania peptydowe. Takie podejście pozwala na szybsze obliczenia sił i oddziaływań, a co za tym idzie możliwość manipulowania większymi obiektami molekularnymi.

Pracownia komputerowa w której prowadzone są zajęcia wyposażona została w system trójwymiarowej wizualizacji zbliżony do technik kinowych 3D, oparty na dwóch rzutnikach posiadających filtry polaryzacyjne. Pracownia wyposażona jest również w stanowiska robocze funkcjonujące w środowisku Scientific Linux, również z możliwością łączenia się z serwerem

Windows'owym. Stanowiska robocze posiadają manipulatory (urządzenia haptyczne, *haptic devices*) firmy SenSable oraz monitory pozwalające na trójwymiarową wizualizację. Wyposażenie pracowni przedstawione jest poniżej.



Rys. 1. Wyposażenie pracowni komputerowej.

Stereoprojektory z polaryzatorami, ekran o powierzchni nie zmieniającej stanu polaryzacji światła, stanowisko robocze z urządzeniem haptycznym i z dwoma monitorami – konwencjonalnym i do wizualizacji 3D (z lewej strony).



Co do oprogramowania, wykorzystywane będą m.in. pakiety:

- NAMD/VMD (<http://www.ks.uiuc.edu/Research/namd/>),
- Visnow ([http://visnow.icm.edu.pl/wiki/index.php/Main\\_Page](http://visnow.icm.edu.pl/wiki/index.php/Main_Page)),
- MOE (Molecular Operating Environment, <http://www.chemcomp.com/>),
- Yasara (<http://www.yasara.org>),
- UNRES (<http://unres.pl/>),
- Mathematica (<http://www.wolfram.com/>) .

Sposób prezentacji struktur krystalograficznych w 3D w publikacjach papierowych zawarty jest m.in. w pracy: B. Lesyng, W. Saenger, *Influence of Crystal Packing Forces on Molecular Structure in 4-thiouridine. Comparison of Anti and Syn Forms*, Z. Naturforsch. **36c**, 956-960 (1981).

## 2. Krótki przegląd stosowanych technologii VR.

Na wykładzie zostanie zaprezentowana technologia kina IMAX oraz wybrane metody animacji filmowych. Informacje na ten temat można znaleźć m.in. w audycjach “Science of the Movies”, patrz m.in. <http://science.discovery.com/tv/science-movies/science-movies.html>

Rys.2 Telewizyjny kanał *Discovery Science* prezentuje serię audycji na ten temat.



Na szczególną uwagę zasługuje największy w świecie i prawdopodobnie najbardziej zaawansowany pod względem technologicznym symulator lotów samolotowych i kosmicznych, patrz:

Rys. 3 (poniżej) oraz [http://news.cnet.com/8301-13772\\_3-10404808-52.html](http://news.cnet.com/8301-13772_3-10404808-52.html)



Wiele z rozwiniętych technik wirtualnej rzeczywistości ze świata makroskopowego, zostało zaadoptowanych do świata molekuł i nanoukładów. Techniki trójwymiarowej wizualizacji obiektów są “w obu światach” bardzo zbliżone. Istotną różnicę stanowią siły. Siły molekularne interpretujemy jako gradienty energii Borna-Oppenheimera, która jest stanem własnym elektronowego równania Schroedingera, przy założeniu że Hamiltonian układu zależy parametrycznie od położenia jąder atomowych. W przybliżonych modelach energii molekularnej stosuje się zazwyczaj analityczne przybliżenie energii B.-O., co zazwyczaj określa się mianem metody mechaniki molekularnej (MM). Jeżeli tak sparametryzowaną energię wykorzystamy do symulacji ruchów atomowych, mamy do czynienia z dynamiką molekularną (MD).

Zakłada się, że uczestnik tego kursu zna i dobrze rozumie metody MM i MD.

## II. Modułowe rozwiązania grafiki komputerowej 2D i 3D oraz wizualnej analizy danych

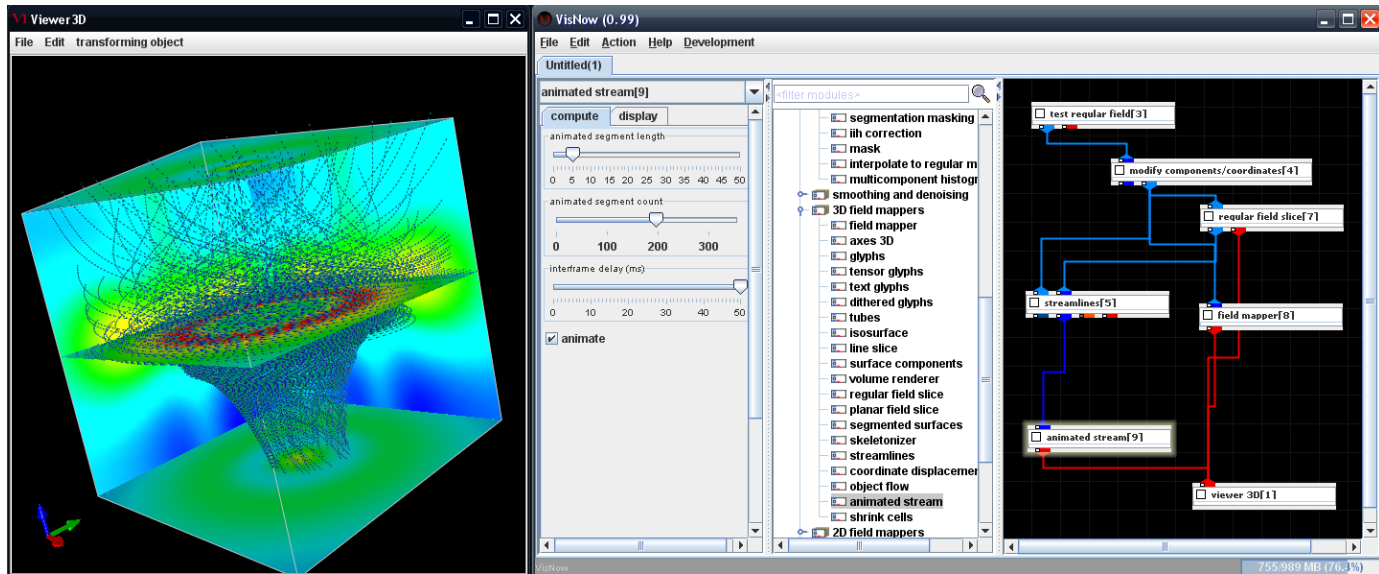
Analiza wizualna to zastosowanie algorytmów i metod przetwarzania i prezentacji wizualnej do analizy danych i wydobywania informacji w nich ukrytej.

Proces postrzegania zmysłowego u człowieka jest w bardzo dużym stopniu przyzwyczajony do ciągłej analizy informacji obrazowej dostarczanej za pośrednictwem narządu wzroku. "Oglądanie" jest dla człowieka naturalnym i intuicyjnym sposobem przetwarzania informacji. Ponadto, przetwarzanie informacji wizualnej charakteryzuje się bardzo znaczącym stopniem upakowania informacji i dużą równoległością przekazu. Konsekwencją powyższych charakterystyk informacji wizualnej jest fakt, iż za pośrednictwem obrazów lub animacji możliwe jest przekazanie bardzo dużej ilości informacji, w krótkim czasie i w intuicyjny dla człowieka sposób.

Z drugiej strony, matematyczne metody analizy danych (np. statystyka) wpisane są często bardzo sztywno w pewne zdefiniowane czy założone ramy - metody te pozwalają znaleźć konkretną ilościową **odповідź na postawione pytanie**, ale pod warunkiem, że potrafimy to pytanie zadać.

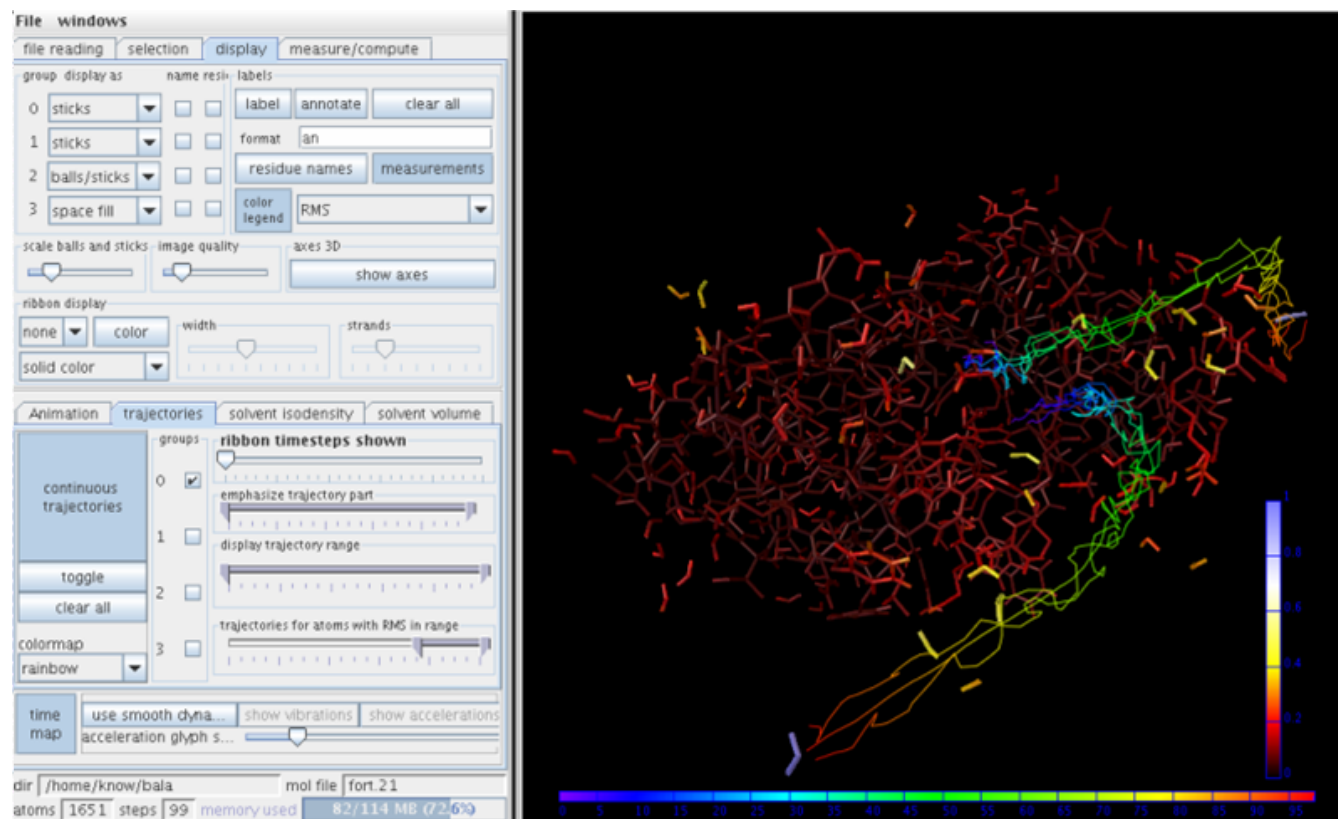
**Analiza wizualna** umożliwia podejście komplementarne - wykorzystując intuicję postrzegania człowieka i możliwości analityczne naszych mózgów, za pośrednictwem obrazów pozwala dostrzec skomplikowane do opisanie matematyczne informacje. Nie wymaga stawiania pytań, a często wręcz **pokazuje nieoczekiwane i pozwala sformułować pytania** oraz poprzeć hipotezy.

Rozmaitość dostępnych danych, które chcemy badać wykorzystując analizę wizualną - od mapy rozkładu ciśnienia na mapie meteo do grafu powiązań semantycznych w artykułach Wikipedii sugeruje wykorzystanie systemów modułowych, składających się z "klocków" przetwarzających dane, zamieniających dane numeryczne na obiekty geometryczne i prezentujących te obiekty. Moduły przekazują między sobą obiekty danych zapisanych w elastycznym, uniwersalnym formacie standardowym. Przykład takiej sieci modułów i wyników jej działania przedstawia poniższy zrzut ekranu. Prezentowany system VisNow jest rozwijany w ICM UW (<http://visnow.icm.edu.pl>).

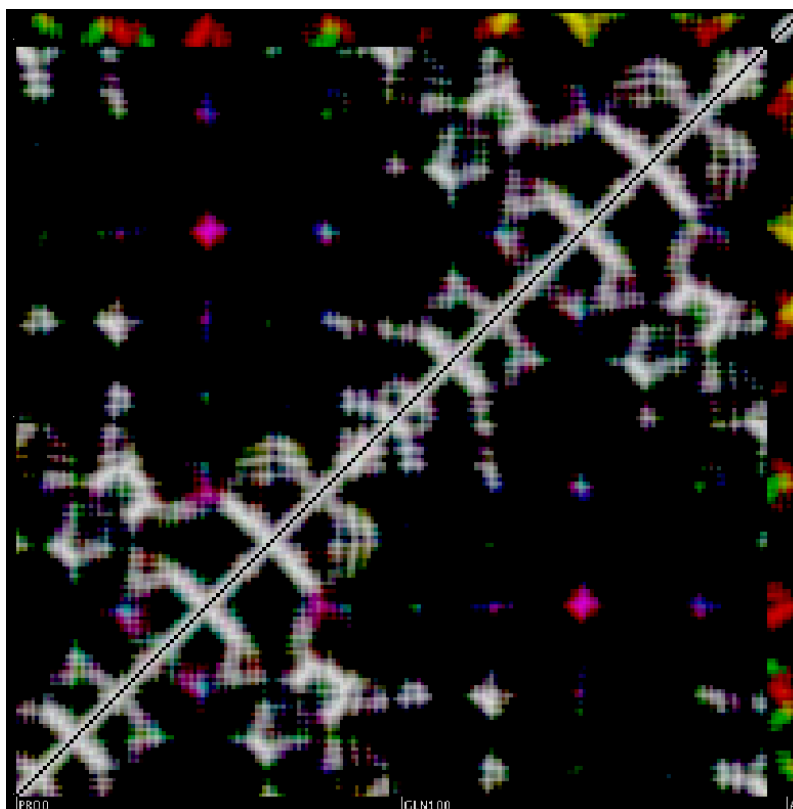


Rys. 1 Wizualizacja trajektorii symulowanego przepływu wirowego

Innym, bardziej wyspecjalizowanym systemem pochodzącym również z ICM UW jest analizator dynamik molekularnych MolDyAna (<http://moldyana.icm.edu.pl>) zawierający szereg niestandardowych metod analizy skomplikowanych przebiegów dynamicznych.



Rys. 2 Obraz systemu molekularnego z trajektoriami wybranych atomów i mapą ruchliwości atomów (dane uzyskane dzięki uprzejmości prof. P. Bały).



Rys. 3 Obraz bliskich kontaktów międzyatomowych w trzech różnych etapach dynamiki molekularnej (dane uzyskane dzięki opreżności prof. J. Trylskiej).

Na wykładach przedstawionych zostanie więcej analiz tego typu.

### *Ćwiczenia*

Na ćwiczeniach studenci otrzymają do analizy wybrane zbiory danych pochodzących z różnego rodzaju symulacji i poproszeni zostaną o samodzielne ich przeanalizowanie.



### III. Programowanie urządzeń haptycznych

#### Wprowadzenie

Manipulatory haptyczne (*haptic devices*) są urządzeniami pozwalającymi użytkownikowi na wskazywanie punktu w układzie współrzędnych związanym z urządzeniem oraz na odczuwanie sił generowanych przez urządzenie. O ile pierwszy aspekt funkcjonalności nie budzi szczególnej wątpliwości i jest naturalnym rozwinięciem urządzeń nie wyposażonych w zwrotną projekcję sił (np. myszy, tabletu graficznego), drugi nie jest oczywisty. Zazwyczaj rozważa się wypadkową siłę działającą na uchwyt urządzenia. Jeżeli urządzenie nie przekazuje żadnej siły, uchwyt powinien sprawiać wrażenie nieważkiego i poruszającego się bez oporów. Jeżeli kursor, który jest reprezentacją pozycji urządzenia w świecie wirtualnym, w swoim ruchu natrafia na obiekt, z którym powinien wejść w interakcję, użytkownik odczuwa to jako opór lub inny przewidziany przez programistę rodzaj siły. Odpowiednio dobierając wartość, kierunek i zwrot siły można tworzyć wrażenie twardości, elastyczności, chropowatości, lepkości. Można również symulować efekty pola, które nie są związane z kontaktem z powierzchnią obiektu, oraz efekty bezwładności i lepkości ośrodka. Innym zastosowaniem urządzeń haptycznych jest ułatwienie użytkownikowi wskazywania określonych punktów lub obszarów poprzez przyciąganie do nich kursora.

Pracownia komputerowa wyposażona jest w urządzenia haptyczne firmy SenSable. Podstawowa dokumentacja oraz przykłady zastosowań zawarte są na stronach <http://www.sensable.com/>.

#### Rodzaje sił

Jak wspomniano powyżej jest wiele sposobów obliczania siły przekazywanej przez urządzenie. Zazwyczaj jest ona funkcją położenia, ruchu lub czasu.

#### Zmienność sił w funkcji położenia lub ruchu

##### *Siła sprężystości*

Siła sprężystości jest najprawdopodobniej najczęściej stosowaną metodą obliczania sił w implementowaniu interfejsów haptycznych. Wynika to z jej prostoty. Wielkość siły sprężystości jest dana prawem Hooke'a ( $F=-kx$ , gdzie  $k$  jest pewną stałą, a  $x$  przemieszczeniem). Wyimaginowana sprężyna jest zaczepiona jednym końcem do pewnego stałego punktu  $p_0$ , drugim zaś do aktualnej lokalizacji kursora urządzenia  $p_1$ . Wektor przemieszczenia  $x=p_1-p_0$  jest tak dobrany, aby siła była zawsze skierowana do punktu  $p_0$ . Siła, którą odczuwa użytkownik jest siłą wywieraną przez wyimaginowaną sprężynę, która usiłuje powrócić do pierwotnej długości wynoszącej 0. Stała  $k$  pozwala regulować sztywność sprężyny

##### *Tłumienie*

Tłumienie w projektowaniu interfejsów haptycznych jest siłą, która przeciwdziała ruchowi. Jest ona proporcjonalna do prędkości z jaką porusza się kursor urządzenia i jest dana równaniem  $F=-bv$ , gdzie  $b$  jest pewną stałą, a  $v$  prędkością z jaką porusza się kursor urządzenia.

### *Tarcie*

Tarcie może być symulowane na wiele sposobów. Najprostsze jest *tarcie kulombowskie*, które jest stałą siłą przeciwdziałającą ruchowi. W przestrzeni jednowymiarowej jest ono dane wzorem  $F = -c \operatorname{sgn}(v)$ . Można je łatwo zaimplementować przy użyciu tłumienia o wysokiej stałej z ograniczeniem na maksymalną wartość siły równym  $c$ . *Tarcie lepkie* implementuje się podobnie, lecz przyjmuje się małą stałą tłumienia i dużą wartość ograniczenia. *Tarcie statyczne i dynamiczne* polegają na przełączaniu się pomiędzy trybem związanym z brakiem ruchu względem powierzchni, o którą następuje tarcie, a trybem związanym z ruchem względem tej powierzchni. Siła tarcia jest proporcjonalna do normalnej składowej siły z jaką użytkownik oddziałuje na powierzchnię i przeciwdziała składowej ruchu równoległej do powierzchni.

### *Bezwładność*

Siła bezwładności jest związana z przyspieszeniem obiektu o niezerowej masie. Jeżeli znane są parametry ruchu można ją wyznaczyć na podstawie drugiego prawa dynamiki.

## **Zmienność sił w funkcji czasu**

### *Siły stałe*

Siła o stałej wartości i kierunku może być zastosowana do równoważenia fizycznej masy ruchomego ramienia urządzenia haptycznego, aby uzyskać jego pozorną nieważkość. Alternatywnie można przy pomocy takiej siły spowodować, że ramię będzie się wydawało cięższe niż jest w rzeczywistości.

### *Okresowe*

Siły okresowe zadane są co do wartości pewną funkcją okresową (np. sinusoidą). Posiada ona ustalony okres i amplitudę. Konieczne jest również określenie kierunku. Amplituda nie może przekroczyć maksymalnej siły jaką urządzenie jest w stanie wytworzyć, w przeciwnym wypadku nastąpi przesterowanie i obcięcie kawałków funkcji określającej wartość siły. Częstotliwość z kolei jest ograniczona częstotliwością komunikacji z urządzeniem haptycznym.

### *Impulsy*

Impuls jest siłą przyłożoną w sposób natychmiastowy. W praktyce impuls trwa pewien krótki czas, zaś siła nie jest przykładana idealnie natychmiastowo. Dąży się do tego, żeby pochodna siły impulsu po czasie była możliwie wysoka. Zauważono bowiem, że dla człowieka nieciągłości w odczuwanej sile są bardzo silnym bodźcem. Mniejsza siła o dużej pochodnej stanowi silniejszy bodziec niż siła większa, ale o mniejszej pochodnej. Należy mieć jednak na uwadze, że konstrukcja urządzenia haptycznego ogranicza raptowność przyłożenia impulsu.

## **Ograniczenia przestrzenne – interakcja z obiektami**

Symulowanie kontaktu (dotyku) z wirtualnymi obiektami sprowadza się do obliczenia i wygenerowania siły, która przeciwdziała wniknięciu punktu związanego z ramieniem urządzenia haptycznego do wewnątrz rozważanego obiektu. Standardową metodą rozwiązania tego problemu jest stosowanie *proxy*, które podąża w wirtualnej przestrzeni za ruchem ramienia urządzenia.

*Proxy* jest zazwyczaj punktem, sferą lub zbiorem punktów. Jeżeli jest ono punktem bywa nazywane punktem kontaktu z powierzchnią (ang. *Surface Contact Point – SCP*). W momencie, gdy ramię urządzenia przenika przez powierzchnię, proxy powinno zostać umieszczone na powierzchni tak, aby zminimalizować energię oddziaływań w układzie powierzchnia-proxy-ramię. Ponadto przemieszczanie się proxy nie może zaburzać spójności przestrzennej. Ostatecznie siła działająca na ramię urządzenia, która będzie powstrzymywała je przed zagłębianiem się do wnętrza obiektu, może być obliczona na podstawie wzajemnego położenia ramienia i proxy jako np. siła sprężystości z tłumieniem. Ta metoda może zostać użyta do odczuwania dowolnego rodzaju kształtów włączając w to analitycznie wyznaczone powierzchnie, siatki wielokątów i bryły voxeli.

### **Integracja projekcji sił z wizualizacją**

Zazwyczaj niewielki sens ma używanie urządzeń haptycznych w oderwaniu od wizualizacji. Zazwyczaj są one stosowane, aby uwiarygodnić i rozszerzyć doznania użytkownika związane z interakcją z wirtualnym trójwymiarowym środowiskiem. Podstawową trudnością w integrowaniu wizualizacji 3D i urządzeń haptycznych jest różnica w częstotliwości odświeżania. Do oszukania zmysłu wzroku i zapewnienia złudzenia ciągłości ruchu wystarczy generowanie zaktualizowanego obrazu z częstotliwością od 30 do 60 razy na sekundę. Z kolei zapewnienie wrażenia stałości siły (np. podczas symulowania kontaktu z twardą powierzchnią) wymaga aktualizowania ok. 1000 razy na sekundę. Podobnie jak przy częstotliwości wyświetlania mniejszej od 30Hz animacja może wydawać się nieciągła i “skacząca”, gdy częstotliwość odświeżania sił spada poniżej 1000Hz użytkownik może odczuwać drżenie i brak precyzji w działaniu urządzenia. Tak duża różnica częstotliwości odświeżania wymusza implementację osobnych wątków odpowiedzialnych za projekcję 3D i projekcję sił. Istotne jest również ograniczenie czasowe związane z częstotliwością odświeżania. W przypadku grafiki i częstotliwości 30Hz na wygenerowanie pojedynczej klatki animacji można poświęcić ok. 33 ms. W przypadku projekcji sił jest to tylko 1 milisekunda, co oznacza, że obliczenia związane z projekcją sił muszą być znacznie prostsze (lub bardziej wydajne) niż te związane z grafiką.

Istnieją biblioteki programistyczne, które przy ustalonym kształcie trójwymiarowej sceny automatycznie tworzą i zarządzają wątkiem odpowiadającym za odświeżanie sił. Jednak w przypadkach, gdy nie mogą być zastosowane (oraz przy ich implementacji), szczególnie ważne jest opracowanie dobrego sposobu aktualizowania informacji o scenie trójwymiarowej. Należy zauważyć, że obydwie wątki (graficzny i haptyczny) muszą mieć dostęp odczytu do aktualnego stanu sceny, oraz mogą go aktualizować. Nie można stosować w tym przypadku tradycyjnej techniki zapewniającej ochronę pamięci współdzielonej, która polega na zabezpieczeniu dostępu przy pomocy semaforów. Mogłoby to doprowadzić do tego, że wątek o wyższym priorytecie (haptyczny) oczekiwałby zbyt długo na wątek graficzny i zaburzona zostałaby częstość odświeżania.

Istotny jest również sposób obsługi zdarzeń. Przykładowo, jeżeli naciśnięcie przycisku na urządzeniu haptycznym ma spowodować pewną zmianę wyświetlanego obrazu oraz zmianę siły odczuwanej

przez użytkownika, nie można oczekiwać na obsłużenie zdarzenia przez wątek graficzny. Reakcja haptyczna musi być natychmiastowa. Należy zatem obsługiwać zdarzenia dwuetapowo. Wątek haptyczny musi niezwłocznie aktualizować siły i kolejkowość część graficzną reakcji na zdarzenie do obsługi przez wątek graficzny przy wyświetlaniu następnej klatki.

### **Integracja projekcji sił z dynamiką modelu**

Interakcja z różnego rodzaju symulacjami dynamiki jest niezwykle naturalnym zastosowaniem urządzeń haptycznych. Pozwala ona użytkownikowi oddziaływać na symulację poprzez wywieranie siły na uczestniczące w niej ciała, zmienianie ich pozycji i nadawanie im pędu oraz odczuwać siły wynikające z oddziaływań występujących w symulacji. Integracja urządzenia haptycznego z symulacją nastręcza jednak wielu trudności. Urządzenia haptyczne mierzą nie siłę jaką użytkownik wywiera na ramię, lecz jedynie położenie ramienia. Siła może być szacowana jedynie poprzez dopasowanie siły równoważącej tak, aby ramię pozostało nieruchome. Techniczne trudności sprawiają, że bezpośrednio przykładanie siły jest niepraktyczne.

Zazwyczaj stosuje się technikę zwaną wirtualnym połączeniem (ang. *virtual coupling*), która wprowadza dodatkową warstwę pomiędzy urządzenie haptyczne, a obiekty symulacji. Zamiast sztywnego połączenia pomiędzy ramieniem urządzenia a symulowanym obiektem, łączy się je sprężyną z tłumieniem, która pośredniczy w przekazywaniu sił. Możliwe jest wtedy w szczególności zastosowanie różnych stałych dla końców sprężyny, co może ułatwiać przekształcanie sił występujących w symulacji na siły przekazywane użytkownikowi. Kolejną trudnością jest częstotliwość odświeżania, która jedynie w przypadku najprostszych symulacji może osiągnąć wielkość umożliwiającą obliczanie kolejnych stanów symulacji w wątku odpowiadającym za komunikację z urządzeniem haptycznym. Zazwyczaj częstotliwość osiągnięta przez interaktywne programy symulacyjne odpowiada o rząd wielkości mniejszej częstotliwości odświeżania obrazu. Oznacza to konieczność wprowadzenia synchronizacji pomiędzy wątkiem symulacji, a wątkiem haptycznym. Technika wirtualnego połączenia przychodzi tutaj z pomocą, ponieważ położenia końców sprężyny mogą być aktualizowane niezależnie przez odpowiednie wątki. Dla zapewnienia gładkości ruchów można stosować techniki inter- i ekstrapolacyjne.

### **Haptyczne interfejsy użytkownika**

Istnieje wiele technik pozwalających na wspomaganie interfejsu użytkownika przy pomocy urządzeń haptycznych. W tym rozdziale przedstawimy najbardziej typowe.

#### **Studnia grawitacyjna**

Typowym scenariuszem użycia jest wybieranie przez użytkownika punktu w przestrzeni trójwymiarowej. Studnia grawitacyjna w znaczący sposób ułatwia wybieranie istniejących w systemie punktów poprzez wprowadzenie sił, które przyciągają ramię urządzenia do punktu, w pobliżu którego się znajduje. Zazwyczaj zasięg działania studni ograniczony jest pewnym promieniem, zaś działająca w jego obrębie siła może być zadana prawem Hooke'a.

### **Studnia grawitacyjna z rzutowaniem**

W przypadku wizualizacji środowiska trójwymiarowego na dwuwymiarowych wyświetlaczach ocena głębokości, na jakiej znajduje się interesujący punkt, może być trudna. W takim przypadku można zmodyfikować metodę studni grawitacyjnej tak, aby ramię urządzenia było przyciągane do punktu jeżeli znajdzie się w pobliżu prostej przechodzącej przez oko obserwatora i rozważany punkt.

### **Operacje niezależne od głębokości**

Rozszerzeniem opisanej powyżej metody jest ignorowanie głębokości podczas manipulowania wybranym obiektem. W tej metodzie podczas operowania obiektem brane są pod uwagę jedynie względne zmiany położenia ramienia urządzenia wzdłuż osi głębokości.

### **Przekształcenia względne**

Urządzenia haptyczne w swojej naturze operują w przestrzeni współrzędnych absolutnych. Aby umożliwić intuicyjne panowanie nad obiektami, należy przekształcać je w związanym z nimi układzie odniesienia. Po wyborze obiektu do transformacji bieżąca pozycja urządzenia haptycznego jest traktowana jako środek układu odniesienia, względem którego mierzony jest obrót i przesunięcie, które zostaną zaaplikowane.

### **Łączenie bodźców wzrokowych z haptycznymi**

Ze względu na psychofizyczne uwarunkowania procesu postrzegania najlepsze efekty uzyskuje się, gdy bodźce wzrokowe są idealnie zgodne z bodźcami dotykowymi. Dlatego niezwykle istotne jest, aby wizualizacja położenia ramienia urządzenia odpowiadała odczuwanym przez użytkownika siłom. Przykładowo, jeżeli użytkownik odczuwa dotyk wirtualnego obiektu, obraz ramienia również powinien się z tym obiektem stykać. Innymi słowy powinien on pokazywać na położenie proxy, a nie samego ramienia, które może być lekko w obiekt zagłębione.

### **Stabilizacja przy pomocy tarcia**

Niejednokrotnie wprowadzenie niewielkiej siły tarcia może być korzystne. Tarcie ułatwia utrzymanie stabilnej pozycji ręki i umożliwia użytkownikowi ustawienie urządzenia we właściwym położeniu. Bez tarcia użytkownik może odczuwać, że ramię porusza się zbyt "luźno", a wykonywanie precyzyjnych ruchów może być utrudnione.

### **Biblioteka HDAPI – niskopoziomowe programowanie interfejsów haptycznych**

Biblioteka *Haptic Device API* (HDAPI) jest częścią pakietu *OpenHaptics Toolkit* opracowanego przez firmę SensAble. Pozwala na komunikowanie się z urządzeniem haptycznym, pobieranie aktualnej pozycji ramienia i ustawianie siły przekazywanej przez urządzenie. Składa się z dwóch zasadniczych komponentów odpowiedzialnych za komunikację z urządzeniem oraz zarządzanie wątkiem obsługującym pętlę odświeżającą siły.

W ramach komunikacji z urządzeniem udostępnione są funkcje służące do jego inicjalizacji, włączania projekcji sił i kalibracji, odpowiadające za zarządzanie stanem i bezpieczeństwem urządzenia (przekroczenia dopuszczalnych sił i prędkości, temperatura silników) oraz pobierania i aktualizacji stanu urządzenia. Stan urządzenia obejmuje współrzędne położenia i prędkość ruchu ramienia jak również macierz obrotu związanego z obrotem uchwytu urządzenia. Aktualizować można siły i

momenty sił przekazywane przez siłowniki. Mogą być one opisane wartościami wektorowymi w kartezyjskim układzie współrzędnych lub parametrami pracy poszczególnych silników.

Moduł zarządzający pętlą odświeżającą siły (*scheduler*) utrzymuje wątek o wysokim priorytecie, który na podstawie pobranego stanu urządzenia oblicza siły i przekazuje je z powrotem. Aktualizacje sił odbywają się z częstotliwością 1000 Hz. Moduł zarządzający umożliwia reszcie aplikacji komunikację z urządzeniem w sposób zapewniający poprawną separację wątków poprzez przekazywanie wywołań zwrotnych (ang. *callbacks*), które wykonywane są w najbliższym kroku pętli aktualizującej stan urządzenia.

Typowy program oparty na bibliotece HDAPI składa się z następujących etapów:

- Inicjalizacja urządzenia.

- Utworzenie wywołania zwrotnego (*callback*), które na oblicza siły na podstawie stanu urządzenia.

- Włączenie projekcji sił.

- Uruchomienie wątku zarządzającego.

- Likwidacja połączenia z urządzeniem i wątku zarządzającego po zakończeniu programu.

Rys. 1: Typowy schemat programu korzystającego z biblioteki HDAPI

**Dostęp do urządzenia haptycznego**

Operacje na urządzeniu haptycznym służą do próbkowania i zmieniania jego stanu i parametrów. Jeżeli uruchomiony jest wątek zarządzający mogą one być wywoływane wyłącznie z poziomu tego wątku. W przeciwnym razie mogłoby nastąpić naruszenie separacji wątków będący jego rezultatem błąd. Można je natomiast wykonywać z dowolnego miejsca aplikacji, jeżeli wątek zarządzający nie jest uruchomiony.

### **Inicjalizacja urządzenia**

Przed pierwszym odwołaniem do urządzenia haptycznego należy je zainicjalizować. Zazwyczaj konfiguracja sterowników urządzenia w systemie operacyjnym wskazuje na urządzenie domyślne. W takim przypadku inicjalizacja urządzenia przyjmuje postać:

```
HHD hHD = hdInitDevice(HD_DEFAULT_DEVICE);
```

Urządzenie po zainicjowaniu ma dla bezpieczeństwa wyłączoną projekcję sił. Aby ją włączyć używa się następującej komendy:

```
hdEnable(HD_FORCE_OUTPUT);
```

Ostatnią komendą, którą wydaje się podczas inicjalizacji jest uruchomienie wątku zarządzającego:

```
hdStartScheduler();
```

Jeżeli aplikacja korzysta z wielu urządzeń każde z nich musi być zainicjalizowane niezależnie, natomiast wątek zarządzający jest jeden na aplikację i powinien być uruchomiony po zainicjalizowaniu wszystkich urządzeń.

Do przełączania się między urządzeniami służy polecenie:

```
hdMakeCurrentDevice(hHD);
```

### **Ramki haptyczne**

Ramki haptyczne określają zakresy, w których stan urządzenia jest niezmienny. W momencie otwarcia ramki (`hdBeginFrame()`) tworzona jest kopia stanu urządzenia. W momencie zamknięcia ramki (`hdEndFrame()`) stan urządzenia jest aktualizowany siłami, które zostały przekazane wewnątrz ramki. Takie podejście gwarantuje, że algorytm obliczający nowe siły na podstawie stanu urządzenia, nie napotka w trakcie obliczeń na zmieniające się dane wejściowe. Przyjmuje się, że na krok pętli zarządzającej urządzeniem powinna przypadać co najwyżej jedna ramka. W przypadku stosowania wielu urządzeń ich ramki są od siebie niezależne. Pobieranie i aktualizowanie stanu urządzenia poza ramką haptyczną może prowadzić do niespodziewanych rezultatów, ponieważ powtarzalność wyniku zapytań o stan nie jest zagwarantowana.

### **Dostęp do stanu urządzenia**

Rodzina funkcji **hdGet** służy do pobierania stanu urządzenia. Funkcje te mają nazwy związane z typem danych, które mają zostać pobrane (np. `hdGetDoublev()`, `hdGetIntegerv()`). Wszystkie wymagają podania stałej określającej rodzaj informacji oraz wskaźnika do komórki pamięci lub tablicy, gdzie ma zostać umieszczony wynik. Zapewnienie przestrzeni wystarczającej do zapisania wyniku spoczywa na aplikacji wywołującej funkcję. Niektóre rodzaje informacji są wspierane tylko przez wybrane funkcje z rodziny **hdGet**. Jeżeli zostanie wywołana funkcja niekompatybilna ze zwracaną wartością nastąpi błąd `HD_INVALID_INPUT_TYPE`. Przykładowo



model urządzenia (stała HD\_DEVICE\_MODEL\_TYPE) jest ciągiem znaków i może zostać pobrany wyłącznie funkcją hdGetString().

Stan bieżący (CURRENT) i poprzedni (LAST) odnoszą się odpowiednio do bieżącej i poprzedniej ramki haptycznej. Pobieranie bieżących wartości wyjściowych (sił – HD\_CURRENT\_FORCE, momentów sił – HD\_CURRENT\_TORQUE i parametrów silników – HD\_CURRENT\_MOTOR\_DAC\_VALUES) zwraca wartości ustawione przez użytkownika w bieżącej ramce. W momencie otwarcia nowej ramki są one automatycznie zerowane. Poniżej przedstawiono przykładowe wywołania funkcji z rodziny **hdGet**:

```
HDint buttonState;
HDstring vendor;
hduVector3Dd position;
HDfloat velocity[3];
HDdouble transform[16];

hdGetInterv(HD_CURRENT_BUTTONS, &buttonState);
hdGetString(HD_DEVICE_VENDOR, vendor);
hdGetDoublev(HD_CURRENT_POSITION, position);
hdGetFloatv(HD_CURRENT_VELOCITY, velocity);
hdGetDoublev(HD_LAST_ENDPOINT_TRANSFORM, transform);
```

Do ustawiania wartości sił służą funkcje z analogicznej rodziny **hdSet**. Poniższy fragment ilustruje ich przykładowe zastosowanie:

```
HDdouble force[3] = {0.5, 0.0, 1.0};
hdSetDoublev(HD_CURRENT_FORCE, force);
HDfloat rampRate = .5;
hdSetFloatv(HD_FORCE_RAMPING_RATE, &rampRate);
```

Należy pamiętać, że ustawione wartości zostaną wysłane do urządzenia dopiero po zamknięciu ramki. Oznacza to w szczególności, że jeżeli na ramieniu urządzenia ma zostać odwzorowana siła będąca wypadkową wielu sił, można wykorzystać wartość HD\_CURRENT\_FORCE do akumulowania tej sumy.

#### **Odzworowywanie sił i momentów sił**

Biblioteka HDAPI wspiera urządzenia haptyczne posiadające możliwość projekcji sił o trzech (tylko siły) i sześciu (siły i momenty sił) stopniach swobody. Istnieje jednak więcej sposobów ustawiania wartości przekazywanych sił wynikających z układu ramion i przegubów w konstrukcji urządzenia. Poniższa tabela ilustruje rodzaje parametrów udostępnianych przez bibliotekę HDAPI:

Rodzaj urządzenia	Parametr	Dotyczy	Układ odniesienia	Jednostki
3DOF/6DOF	HD_CURRENT_FORCE	Siły	Kartezjański	N (F_x, F_y, F_z)

Rodzaj urządzenia	Parametr	Dotyczy	Układ odniesienia	Jednostki
6DOF	HD_CURRENT_TORQUE	Momenty sił	Kartezjański	mNm ( $T_x, T_y, T_z$ )
3DOF/6DOF	HD_CURRENT_MOTOR_DAC_VALUES	Siły i momenty sił	Przestrzeń przegubów	od -32768 to 32768 ( $M_1, M_2, M_3, \dots$ )
3DOF/6DOF	HD_CURRENT_JOINT_TORQUE	Przeguby ramienia	Przestrzeń przegubów	mNm ( $T_{J1}, T_{J2}, T_{J3}$ )
6DOF	HD_CURRENT_GIMBAL_TORQUE	Przeguby uchwytu	Przestrzeń przegubów	mNm ( $T_{J4}, T_{J5}, T_{J6}$ )

Należy pamiętać, że parametr HD\_CURRENT\_MOTOR\_DAC\_VALUES ma pierwszeństwo przed HD\_CURRENT\_JOINT\_TORQUE i HD\_CURRENT\_GIMBAL\_TORQUE, które z kolei mają pierwszeństwo przed HD\_CURRENT\_FORCE i HD\_CURRENT\_TORQUE.

## Przykłady i ćwiczenia:

```
HDfloat gimbalTorque[3];
gimbalTorque[0] = torque_yaw;
gimbalTorque[1] = torque_pitch;
gimbalTorque[2] = torque_roll;
hdSetFloatv(HD_CURRENT_GIMBAL_TORQUE, torque);
HDdouble baseTorque[3] = {100, 250, 200};
//Base Torque in mNm
hdSetDoublev(HD_CURRENT_JOINT_TORQUE, baseTorque);
HDdouble gimbalTorque[3] = {30, 65, 0.0};
//Gimbal Torque in mNm
hdSetDoublev(HD_CURRENT_GIMBAL_TORQUE, gimbalTorque);
```

### Praca z wątkiem zarządzającym, rejestrowanie wywołań zwrotnych

Jak wspomniano powyżej wątek zarządzający wykonuje z dużą (1000 Hz) częstotliwością pętlę, w której wykonywane są zarejestrowane przez użytkownika wywołania zwrotne. Wszystkie odwołania do stanu urządzenia i zmiany sił powinny być realizowane za pośrednictwem odwołań zwrotnych, ponieważ daje to gwarancję, że pobrane i wysłane informacje będą spójne (będą dotyczyły jednej ramki haptycznej).

Deklaracja funkcji będącej wywołaniem zwrotnym ma następującą postać:

```
HDCallbackCode HDCALLBACK DeviceStateCallback (void *pUserData);
```

Wartość zwracana przez wywołanie zwrotne informuje wątek zarządzający, czy było ono jednorazowe (HD\_CALLBACK\_DONE). Jeżeli ma ono zostać powtórzone w następnym kroku pętli powinna zostać zwrócona wartość HD\_CALLBACK\_CONTINUE. Przykładowo, jeżeli celem jest jednorazowe pobranie położenia i sił, można zastosować poniższą funkcję:

```
struct DeviceDisplayState {
    HDdouble position[3];
    HDdouble force[3];
}

HDCallbackCode HDCALLBACK DeviceStateCallback (void *pUserData)
{
    DeviceDisplayState *pDisplayState =
        static_cast<DeviceDisplayState *>(pUserData);
    hdGetDoublev(HD_CURRENT_POSITION, pDisplayState->position);
    hdGetDoublev(HD_CURRENT_FORCE, pDisplayState->force);
```

```

        return HD_CALLBACK_DONE;
    }

```

Tak zdefiniowaną funkcję należy zgłosić do wykonania przez wątek zarządzający. Biblioteka HDAPI implementuje dwa tryby rejestrowania wywołań zwrotnych: synchroniczny i asynchroniczny. W trybie synchronicznym wywołanie funkcji zgłaszające wywołanie zwrotne kończy się dopiero wtedy, gdy zostanie ono wykonane. Tryb ten należy stosować w przypadku wywołań, których celem jest odczytanie stanu urządzenia. Funkcję `DeviceStateCallback` można zarejestrować w następujący sposób:

```

DeviceDisplayState state;
hdScheduleSynchronous(DeviceStateCallback,
                       &state,
                       HD_MIN_SCHEDULER_PRIORITY);

```

Tryb asynchroniczny jest właściwy przy rejestrowaniu wywołań, które mają obliczać siły, które będą aktualizowane przez pętlę wątku zarządzającego. Poniższy przykład demonstruje jak można zdefiniować funkcję odpowiadającą za generowanie siły:

```

HDCallbackCode HDCALLBACK CoulombCallback(void *data)
{
    HHD hHD = hdGetCurrentDevice();
    hdBeginFrame(hHD);
    HDdouble pos[3];
    //Pobieranie pozycji ramienia
    hdGetDoublev(HD_CURRENT_POSITION, pos);
    HDdouble force[3];
    // Obliczanie siły
    forceField(pos, force);
    // Ustawianie siły w urządzeniu
    hdSetDoublev(HD_CURRENT_FORCE, force);
    // Wysyłanie siły.
    hdEndFrame(hHD);
    // Wywołanie ma być wykonane w następnym kroku pętli
    return HD_CALLBACK_CONTINUE;
}

```

```

hdScheduleAsynchronous(AForceSettingCallback, (void*)0,
                       HD_DEFAULT_SCHEDULER_PRIORITY);

```

## **Zakończenie pracy**

Po zakończeniu korzystania z urządzenia haptycznego należy zatrzymać wątek zarządzający, usunąć zarejestrowane wywołania zwrotne oraz zakończyć pracę z urządzeniem. Poniżej przedstawiono typowy zestaw instrukcji wykonujących te czynności:

```
hdStopScheduler();  
hdUnschedule(scheduleCallbackHandle);  
hdDisableDevice(hdGetCurrentDevice());
```

#### **IV. Zastosowania manipulatorów w modelowaniu prostych układów biomolekularnych i nanoukładów w środowisku NAMD/VMD**

##### *Wymagania wstępne:*

Studenci muszą znać podstawy obsługi systemu Linux oraz znać podstawy teoretyczne metod mechaniki i dynamiki molekularnej. Komputery muszą posiadać zainstalowane i skonfigurowane manipulatory, oraz oprogramowanie: VMD, NAMD, VRPN – dostępne dla studentów. W katalogach roboczych muszą zostać umieszczone katalogi z ćwiczeniami: „nanotube” i „ismd”. Podstawy technik symulacyjnych z wykorzystaniem urządzeń haptycznych (haptic devices), zawarte są m.in. w publikacjach [1-3].

##### **Wstęp**

Symulacje dynamiki molekularnej (MD) pozwalają nam na badanie zachowania układów molekularnych, np. zmian konformacyjnych struktury natywnej białka, w określonym środowisku. Zazwyczaj odbywa się to w następujący sposób: wybrany układ zostaje przygotowany do symulacji, obliczenia zostają uruchomione i dopiero po zakończeniu symulacji możemy oglądać jej rezultaty, przygotowując układ z nowymi parametrami do kolejnej symulacji.

Sterowana dynamika molekularna (SMD) jest modyfikacją zwykłej dynamiki molekularnej polegającą na wprowadzeniu sił zewnętrznych działających na wybrany atom lub grupę atomów. Umożliwia to wymuszenie ruchu atomów lub cząsteczek w wybranym przez nas kierunku pozwalając na obserwację zjawisk takich jak transport ligandów do miejsca aktywnego receptora lub mechaniczne rozfałdowywanie białek.

Sprzęgnięcie programów do wizualizacji układów molekularnych i trajektorii MD pozwala na oglądanie na bieżąco zmian w symulowanym układzie. Po uwzględnieniu metod SMD dostajemy interaktywną dynamikę molekularną (IMD) umożliwiającą obserwację trajektorii na bieżąco oraz zaburzenie układu poprzez wprowadzenie dodatkowych sił. Komplet programów typu VMD/NAMD pozwala na wprowadzanie takich zmian na ekranie programu wizualizacyjnego poprzez wybór atomu, kierunku działania oraz wartości siły przy pomocy myszki. Po odpowiednim skonfigurowaniu programu do IMD umożliwiają na współpracę ze specjalnymi manipulatorami pozwalając na dwustronne przesyłanie informacji o oddziałujących siłach – użytkownik może sam poczuć opór układu molekularnego przy wprowadzaniu dodatkowych sił.

W rezultacie otrzymujemy w pełni dopracowane interaktywne środowisko. Praca w nim odbiega od znanych do tej pory rozwiązań stosowanych w pracowniach teoretycznych. Celem poniższych ćwiczeń jest zapoznanie studentów z możliwościami takich rozwiązań oraz oswojenie się ze stosowanymi urządzeniami.

## Ćwiczenia:

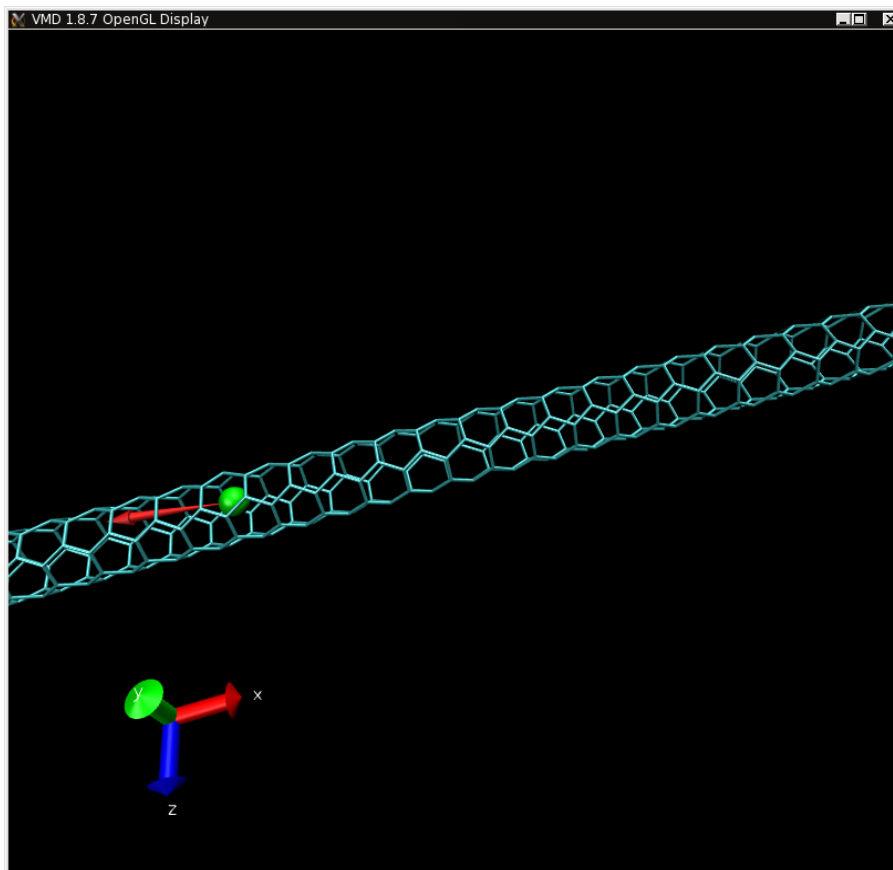
(Linie zaczynające się od # są poleceniami konsoli, które należy wykonać)

Przed przystąpieniem do ćwiczeń przeprowadzamy test manipulatorów uruchamiając aplikację diagnostyczną:

```
# strace /usr/sbin/PHANToMTest
```

## Ćwiczenie 1

### Nanorurka



Rys 1. Okno programu VMD z nanorurką i atomem żelaza (zielona kulka) w środku. Czerwoną strzałką wychodzącą z atomu żelaza oznaczony jest wektor przyłożonej siły. W lewym dolnym rogu podane są wektory jednostkowe kartezjańskiego układu współrzędnych.

Jak uruchomić symulację znajdującą się w tym katalogu (np. /home/agorecki/haptic-demo/nanotube) ?

1) Uruchamiamy w terminalu program dynamiki molekularnej NAMD z odpowiednim plikiem wejściowym \*.namd:

```
# cd /home/agorecki/haptic-demo/nanotube
# namd2 nanotube.namd
```

Program uruchamia się i zaczyna nasłuchiwać połączeń na porcie TCP 6666.

Info: *Interactive MD listening on port 6666.*

Info: *Startup phase 7 took 0.00406004 s, 2.48783 MB of memory in use*

Info: *Startup phase 8 took 0.000139243 s, 3.61442 MB of memory in use*

Info: *Finished startup at 6288.6 s, 3.61442 MB of memory in use*

TCL: *Minimizing for 100 steps*

Info: *INTERACTIVE MD AWAITING CONNECTION*

Symulacja jest wstrzymana do czasu, aż nastąpi połączenie na porcie 6666 z programu do wizualizacji VMD.

2) W oddzielnym terminalu uruchamiamy program serwera haptic devices

```
# vrpn
```

3) Uruchamiamy program wizualizacyjny VMD

```
# cd /home/agorecki/haptic-demo/nanotube
# vmd -e nanotube.vmd
```

Po kilku sekundach pojawia się wizualizacja nanorurki z zielonym atomem w środku.

Program VMD podłącza się do portu 6666 na tej samej maszynie.

Od tego czasu można przykładać siły do wybranych atomów przy pomocy urządzenia *haptic-device*.

Przykłady:

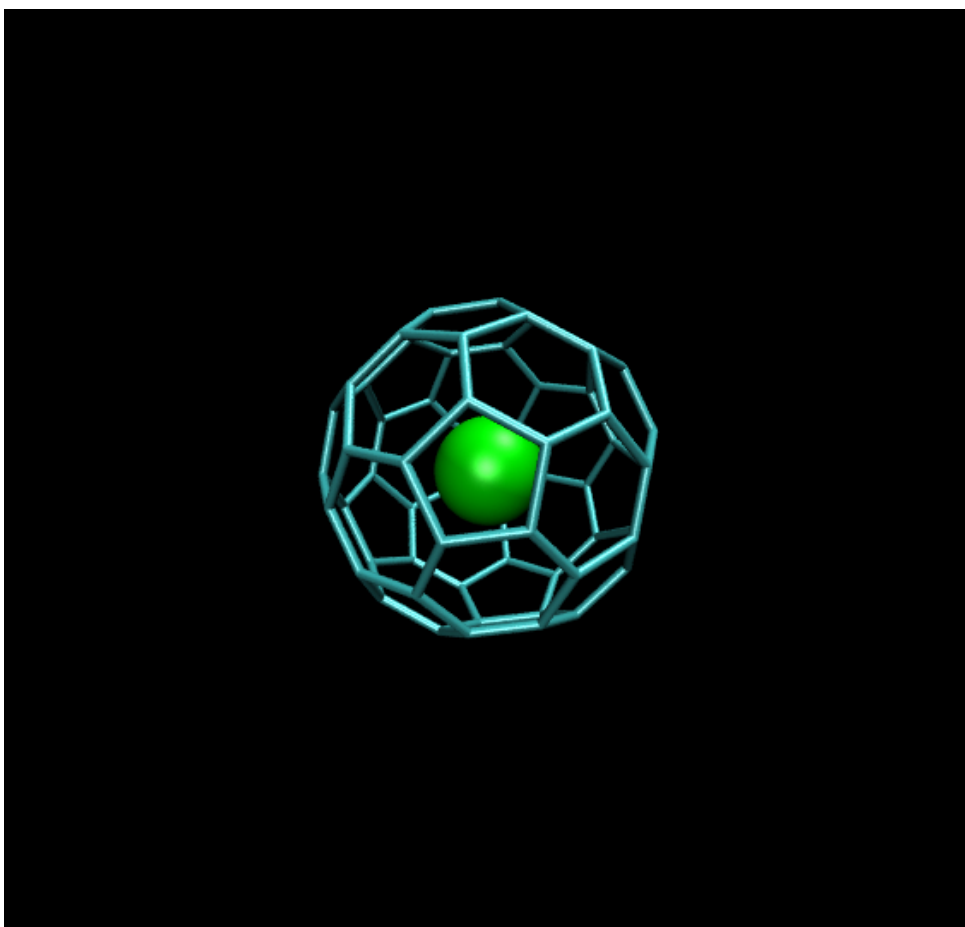
+ próba przesuwania zielonego atomu wzdłuż rurki

+ próba wyjęcia atomu przez oczko sieci



## Ćwiczenie 2

### Fuleren



Rys 2. Fuleren C<sub>60</sub> z umieszczonym w środku atomem żelaza.

W celu uruchomienia symulacji wykorzystujemy pliki konfiguracyjne z katalogu „fullerene” w sposób analogiczny do poprzedniego ćwiczenia:

fullerene.namd - plik konfiguracyjny do programu NAMD

fullerene.vmd - plik konfiguracyjny do programu VMD

Po uruchomieniu VMD powinniśmy mieć przed oczyma okno VMD OpenGL Display z reprezentacją fulerenu i atomu żelaza jak na zaprezentowanym rysunku.

Ćwiczenie polega na próbie wyjęcia atomu przez oczko cząsteczki C<sub>60</sub>. Należy zwrócić uwagę na różnicę zachowania nanorurki z poprzedniego ćwiczenia i cząsteczki fulerenu. Co jest odpowiedzialne za te różnice?

### Ćwiczenie 3

#### Niewielkie białko



Rys 3: Konektyna, białko mięśni poprzecznie prążkowanych kręgowców, składa się głównie z beta-kartek (żółte fragmenty). Końce białka zostały wyróżnione kolorowymi kulami: N-koniec – czerwoną, C-koniec – niebieską

Postępujemy analogicznie jak w ćwiczeniach poprzednich. Katalogiem roboczym jest katalog „ismd” gdzie znajdują się pliki:

connectin.conf – plik konfiguracyjny do programu NAMD,

connectin.vmd – plik konfiguracyjny do programu VMD.

Po uruchomieniu VMD powinniśmy mieć przed oczyma okno VMD OpenGL Display z reprezentacją białka z wyróżnionymi helisami i beta-kartkami – dokładnie taką jak na rysunku powyżej.

Ćwiczenie polega na znalezieniu najslabiej oddziałujących elementów struktury trzeciorzędowej białka poprzez mechaniczne rozfałdowywanie poszczególnych fragmentów. Przy odpowiednio niewielkiej deformacji struktury można zaobserwować powrót do struktury natywnej.

Dla grup zaawansowanych i zainteresowanych proponowane jest rozszerzenie ćwiczenia 3:

Ćwiczenie należy powtórzyć dla symulacji z plikiem konfiguracyjnym programu NAMD – connectin1.conf oraz wskazać przy których parametrach symulacji ćwiczenie jest łatwiejsze do wykonania.

#### *Literatura:*

1. W. Humphrey, A. Dalke, K. Schulten, *VMD: visual molecular dynamics*. J Mol Graph. **14**, 33-8, 27-8 (1996)
2. J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, K. Schulten, *Scalable molecular dynamics with NAMD*, J Comput Chem. **26**, 1781-802 (2005)
3. H. Lu, B. Isralewitz, A. Krammer, V. Vogel, K. Schulten, *Unfolding of titin immunoglobulin domains by steered molecular dynamics simulation*, Biophys J. **75**, 662-71 (1998)

## V. Wizualizacja 3D wyników symulacji metodami dynamiki molekularnej złożonych układów biomolekularnych.

### Wstęp

#### 1. Jak widzi ludzkie oko

Czynniki umożliwiające widzenie stereoskopowe

- psychologiczne - skutek nabytego doświadczenia:
  - wielkość znanych obiektów,
  - perspektywa liniowa,
  - perspektywa atmosferyczna,
  - przesłanianie,
  - rozkład światła i cieni,
  - paralaksa ruchu,
- fizjologiczne - wynikające z budowy narządu wzroku:
  - akomodacja,
  - zbieżność,
  - nierównoległość siatkowa.

Mechanizm widzenia i podstawowe pojęcia z nim związane.

*Ćwiczenie 1.* Kilka zadań na wykrywanie oka dominującego.

#### 2. Przypomnienie historii rozwoju obrazowania 3D

- Euklides -prawa optyki- 300 lat p.n.e.
- Gimbattista della Porta - widzenie przestrzenne - XVI w.
- Charles Wheatstone - stereoskop - 1838
- Ducos du Hauron - technika anaglifowa - 1890
- premiera filmu 3D „The Power of Lowe” - 27.09.1922
- Edwin H.Land - film polaryzacyjny -1929
- B-klasowe filmy -lata 50
- 3D w IMAX-ach - 1986
- film „Awatar” -2009

### 3. Stereogramy

*Side-by-side* czyli tak zwana stereopara oznacza, że mamy dwa obrazki. Na każdym z nich jest ten sam widok/rysunek z nieco innej perspektywy, przystosowany specjalnie dla konkretnego oka. W zależności od układu tych rysunków (czyli gdy obrazek po stronie lewej jest przeznaczony dla lewego oka a a po stronie prawej dla prawego lub gdy obrazek po stronie lewej jest przystosowany do oka prawego, a po stronie prawej do oka lewego) musimy przystosować nasze patrzenie tak aby zobaczyć obraz w trójwymiarze. I tak:

- gdy obrazek po stronie lewej jest przeznaczony dla lewego oka a a po stronie prawej dla prawego , patrzymy równolegle na oba obrazki (ang. *Side-by-side wall-eyed stereo*), i aby widzieć w 3D musimy zrobić zeza rozbieżnego;
- gdy obrazek po stronie lewej jest przystosowany do oka prawego, a po stronie prawej do oka lewego, patrzymy krzyżowo na oba obrazki (ang. *Side-by-side cross-eyed stereo*), czyli musimy zrobić zeza zbieżnego.

Stereogram *random dot*, jak sama nazwa wskazuje, wygląda jak obraz składający się z losowo rozmieszczonych kolorowych kropek. Aby zobaczyć 3D w tego typu obrazie musimy zrobić zeza rozbieżnego. To co zobaczymy może nas trochę rozczarować. Choć obraz jest kolorowy, to obraz w 3D jest tylko półprzezroczysty, ale z bardzo wyraźnymi konturami ukazanego przedmiotu/napisu. Wrażenie dużej głębi uzyskujemy, przez zastosowanie pełnej gamy kolorów dla naszych kropek.

*Anglif stereo* składa się z dwóch obrazów, każdy z nich przystosowany jest dla konkretnego oka, nałożonych na siebie. Ponieważ lewe i prawe oko widzi tą samą rzecz pod nieco innym kontem, anglif wygląda jak nieostre zdjęcie. Aby to zdjęcie zobaczyć w 3D musimy mieć specjalne okulary, w których jedno szkło jest czerwone, a drugie niebieskie. Nałożone na siebie obrazy preferują barwę czerwoną lub niebieską, dzięki temu po nałożeniu okularów, każde oko widzi inną fotografię, a tym samym mamy wrażenie obrazu w 3D. Niestety kolorystyka obrazu jest szarobura.

- Stworzenie anaglifu w Photoshopie

Oznaczmy składowe barw obrazu dla oka lewego poprzez:  $L_r$  (składowa czerwona),  $L_g$  (składowa zielona),  $L_b$  (składowa niebieska), analogicznie obraz dla oka prawego  $R_r$ ,  $R_g$ ,  $R_b$  oraz anglif czyli wynikowy obraz trójwymiarowy  $A_r$ ,  $A_g$ ,  $A_b$ .

- Tworzenie anaglifu monochromatycznego:

Zamienimy obrazy na odcienie szarości lecz w obrazie oka lewego pokażemy tylko składową barwy czerwonej, natomiast w obrazie oka prawego składowe barwy zielonej i niebieskiej. Do przekształcenia obrazu kolorowego w monochromatyczny użyjemy wzoru na luminancję w modelach barw używany w telewizji.

Dla obrazu oka lewego:

$$\text{składowa czerwona} = 0.299 * L_r + 0.587 * L_g + 0.114 * L_b$$

$$\text{składowa zielona} = 0$$

$$\text{składowa niebieska} = 0$$

Dla obrazu oka prawego:

$$\text{składowa czerwona} = 0$$

$$\text{składowa zielona} = 0.299 * R_r + 0.587 * R_g + 0.114 * R_b$$

$$\text{składowa niebieska} = 0.299 * R_r + 0.587 * R_g + 0.114 * R_b$$

Otrzymujemy dwa obrazki, by uzyskać anaglif musimy je połączyć w jeden - czyli dodania do siebie obu obrazów dla oka lewego i prawego. Zatem przekształcenie obu obrazów do anaglifu monochromatycznego możemy przedstawić następująco:

$$A_r = 0.299 * L_r + 0.587 * L_g + 0.114 * L_b$$

$$A_g = 0.299 * R_r + 0.587 * R_g + 0.114 * R_b$$

$$A_b = 0.299 * R_r + 0.587 * R_g + 0.114 * R_b$$

- Tworzenie anaglifu - algorytm Photoshop:

Algorytm ten pozwala uzyskać anaglif kolorowy. Jego działanie jest bardzo proste. Do wynikowego obrazu bierzemy składową barwy czerwonej z obrazu dla oka lewego oraz składowe zieloną i niebieską z obrazu dla oka prawego. Całą operację można zapisać następującym wzorem:

$$A_r = L_r$$

$$A_g = R_g$$

$$A_b = R_b$$

Wadą tego algorytmu jest branie wyłącznie składowej barwy czerwonej z obrazu dla oka lewego. Dla niektórych obrazów - takich które zawierają mało tej składowej efekt może być niezadowalający. Możemy to zmodyfikować przekształcając obraz dla oka lewego do odcieni szarości przed zastosowaniem tego algorytmu. Wówczas wzór ulega modyfikacji do następującej postaci:

$$A_r = 0.299 * L_r + 0.587 * L_g + 0.114 * L_b$$

$$A_g = R_g$$

$$A_b = R_b$$

- Stereogram z przeplotem do tego typu stereogramu potrzebne są ciekłokrystaliczne okulary migawkowe, dobra karta graficzna w komputerze i monitor z wysoką częstotliwością odświeżania (najczęściej monitory kineskopowe, choć wybrane modele monitorów LCD również się nadadzą). Oczywiście wszystkie elementy muszą być ze sobą zsynchronizowane. Zasada działania jest następująca: okulary kilkadziesiąt razy na sekundę zaciemniają raz jedno oko, raz drugie, zsynchronizowany z nimi komputer wyświetla, w zależności od nie zaciemnionego oka, obraz raz dla lewego, raz dla prawego oka, dzięki temu wydaje nam się, że widzimy obraz w 3D. Obraz jest w pełni realistyczny i bez ograniczeń kolorystycznych. W najnowszych programach do modelowania ten typ stereo jest obecnie wykorzystywany.
- Stereoskopia polaryzacyjna do tego typu stereogramu potrzebujemy okulary (inne niż opisane wyżej, dobrą karta graficzna w komputerze, dwa rzutniki z filtrami polaryzacyjnymi o prostopadłych względem siebie kierunkach polaryzacji światła, specjalny ekran. Przez okulary, które są spolaryzowane w taki sam sposób jak rzutniki, do każdego oka dociera obraz przystosowany specjalnie dla tego oka. Dzięki temu mamy wrażenie obrazu w 3D o naprawdę wysokiej jakości, możemy go oglądać nawet pod kątem.

#### 4. Techniki 3D w programach do dynamiki molekularnej

##### **Opis i możliwości programu *Molecular Operating Environment (MOE) Chemical Computing Group*.**

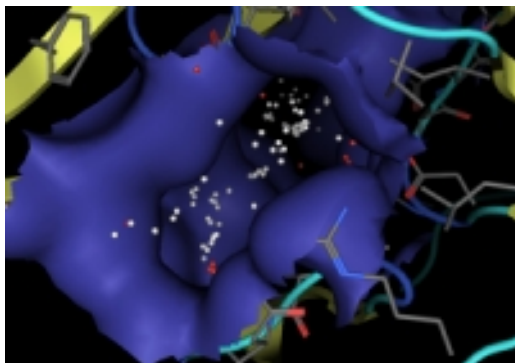
Jest to zintegrowany system do projektowania leków. Pozawala on na stworzenie modelu białka (ma wbudowane pakiety do tworzenia modelu homologicznego, ale również do tworzenia modelu metodą przewlekania, posiada wygodny interfejs do przeszukiwania strukturalnych baz danych takich jak PDB), jego optymalizację (narzędzia do mechaniki i dynamiki molekularnej), zadokowanie do modelu znanego liganda (znajdowanie miejsca aktywnego, dokowanie) jak i zaprojektowanie własnego (generator ligandów 3D, narzędzia do projektowania farmakoforów). Wszelkie informacje o tym programie znajdują się na stronie internetowej: <http://www.chemcomp.com/software.htm>. Charakterystyka i omówienie funkcjonalności następujących narzędzi z programu MOE:

- przeszukiwania strukturalnych baz danych takich jak PDB, modelowanie homologiczne, modelowanie metodą przewlekania, określenie protonacji białka,
- znajdowanie i wizualizacja miejsca aktywnego w białku, analiza oddziaływania kompleksu białko-ligand, generator ligandów 3D, dokowanie ligandu do białka i jego analiza, optymalizacja ligandu w miejscu aktywnym, *Multi-Fragment Search*,

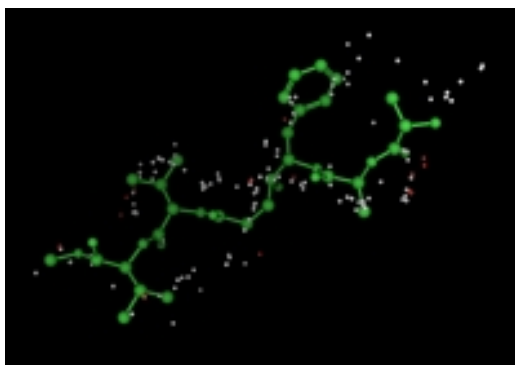
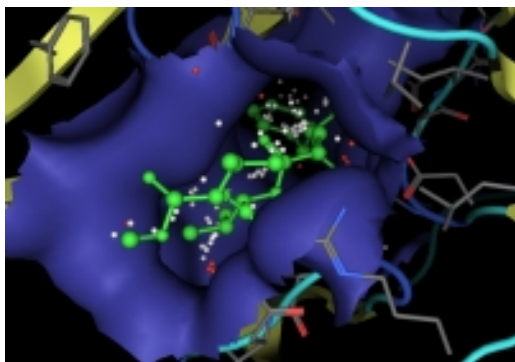
tworzenie map kontaktów, interakcji czy elektrostatycznych kompleksu białko-ligand, projektowanie farmakoforów.

Zdjęcia zostały pobrane ze strony CCG <http://www.chemcomp.com/journal/sitefind.htm>

Zdjęcia poniżej pokazują przykład użycia narzędzia do znajdowania miejsca aktywnego w białku. Białko to struktura z bazy danych PDB - 1AAQ (proteaza HIV-1). Pierwsze zdjęcie pokazuje sfery alfa, za



pomocą których poszukiwane jest miejsce aktywne. Drugie zdjęcie pokazuje miejsce aktywne z ligandem (*hydroxyethylene isostere*). Trzecie - pokazuje pokrycie alfa sfer przez ligand.



Zostaną również omówione dostępne w programie funkcje 3D z ze szczególnym uwzględnieniem trybu stereo.



Na stronie internetowej znajdują się linki do publikacji CCG z poszczególnych działów, z których również będą korzystali studenci.

**Ćwiczenie 2.** Zostanie stworzony model kinazy LCK za pomocą narzędzi programu MOE (model kinazy LCK w PDB nie jest kompletny, więc zostanie zastosowane modelowanie homologiczne i przeszukiwanie bazy PDB), następnie zostanie określona protonacja wymodelowanego białka. Do tak przygotowanej struktury zostanie zadokowany ligand. A otrzymane wyniki poddane zostaną analizie. Całe ćwiczenie zostanie wykonane w trybie stereo.

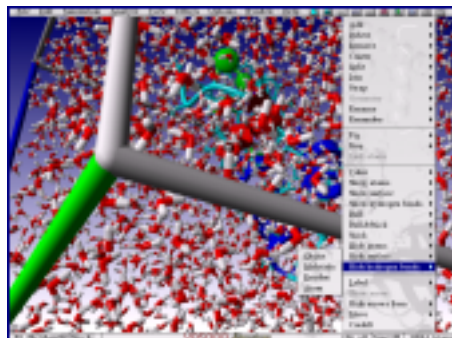
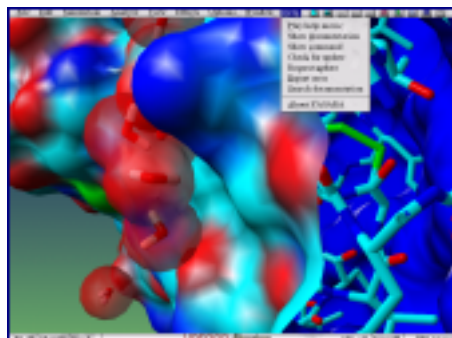
Dzięki współpracy z firmą *Chemical Computing Group* jest pełny dostęp do tutoriali stworzonych przez pracowników tej firmy, w formie elektronicznej.

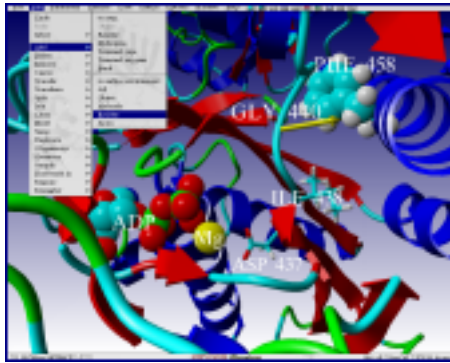
### **Opis i możliwości programu Yasara (*Yet Another Scientific Artificial Reality Application*)**

<http://www.yasara.org>.

Yasara jest to program do dynamiki i mechaniki molekularnej, z intuicyjnym panelem graficznym do rysowania cząsteczek chemicznych, oraz przedstawianiem ich w różnych reprezentacjach. Posiada funkcje 3D, które zostaną omówione.

Oto przykładowe obrazy z Yasary:





W Yasarze mamy możliwość tworzenia własnych prezentacji i przedstawiania ich w trybie stereo. Oprócz tego pozwala w tych prezentacjach na wykonywanie obliczeń MM i MD w czasie rzeczywistym, w formie pasywnej i interaktywnej (mamy możliwość zakłócenia trwających obliczeń DM przez przyłożenie siły do atomu lub grupy atomów). Na wykładzie skupimy się na analizie kodu dla prezentacji 3D.

Oto przykładowy fragment kodu dostępny w programie

Yasara:

Obrazy pochodzą ze strony internetowej <http://www.yasara.org>

```
# YASARA MOVIE
# TOPIC:      3. Molecular Dynamics
# TITLE:      3.1. The theory behind molecular dynamics simulations
# REQUIRES: Dynamics
# AUTHOR:     Elmar Krieger
# LICENSE:    GPL

Help=1
if Help
    # If we are showing a help movie, wait for clicks on the
ContinueButton
    Button='ContinueButton'
else
    # Otherwise just wait for a click on the left mouse button
    Button='LeftButton'

# Change working directory to the place of this macro
# (so that we don't have to specify paths when loading stuff).
CD (MacroDir)
RequireVersion 3.11.16
```

```

OnError Continue
UserInput Off
Console Off

Boundary Wall
Longrange None

#Menu Off

#
# SLIDE 1: Title Page
#
TitlePage:
# Left handed coordinate system
Clear
CoordSys Left
Projection perspective
HideMessage
LightSource 45,45,Ambience=85,Shadow=65
AtomSize 100

# Don't show console and head up display
HUD Off
MarkAtom None
StickRadius 70
BallStickRadius 50,70

Style Ball
FontFog Off
ColorBG Blue,White
# The rotating protein
LoadPDB gprotein
PosObj gprotein,14.7,-11,17
StickObj gprotein
ColorObj gprotein,Cyan
AutoRotateObj gprotein,Y=-0.1
# The title
MakeTextObj Title,Width=800,Height=80
Font
Name=Baikal,Height=40%,Color=White,Spacing=2,Depth=100,DepthCol=ff80
00

```

```

PosText 50%,50%,Justify=Center
Print "Molecular Dynamics Simulations\n"
Print "Watching Nature@Work"
PosObj Title,0,190,730
MakeTextObj Author,Width=20,Height=1
Font Name=Baikal,Height=100%,Color=Black,Depth=0
PosText 50%,100%
Print "by\n"
Print "Elmar.Krieger@cmbi.ru.nl\n"
PosObj Author,0,-3,18.9
# Load Logos
LoadPNG logos
ShowImage logos,0,0,1024,768,Alpha=100,Priority=1
if Help
    ShowMessage 'Hi (FirstName)! Welcome to a 15-minute YASARA Movie
about the theory of molecular dynamics simulations.'
Wait 1
# Take snapshot here, Eliza!
# MovieSnapshot
Wait (Button)
if Help
    ShowMessage "Imagine you had a microscope with almost infinite
magnification, like those guys on starship Voyager..."
AutoMoveObj gprotein,X=0.1
Wait 20
AutoRotateObj Author,X=-5
AutoMoveObj Author,Y=-0.05,Z=-0.1
Wait 20
# Look through 'o'
AutoMoveToObj Title,-204.7,52.4,50.5,Steps=300
DelObj Author
# Show ProteinG again
AutoMoveObj gprotein,X=0
AutoRotateObj gprotein,Y=0
OriObj gprotein,0,0,0
ColorObj gprotein,Element
BallObj gprotein
PosObj gprotein,33,0,278
Wait (Button)
if Help

```

```

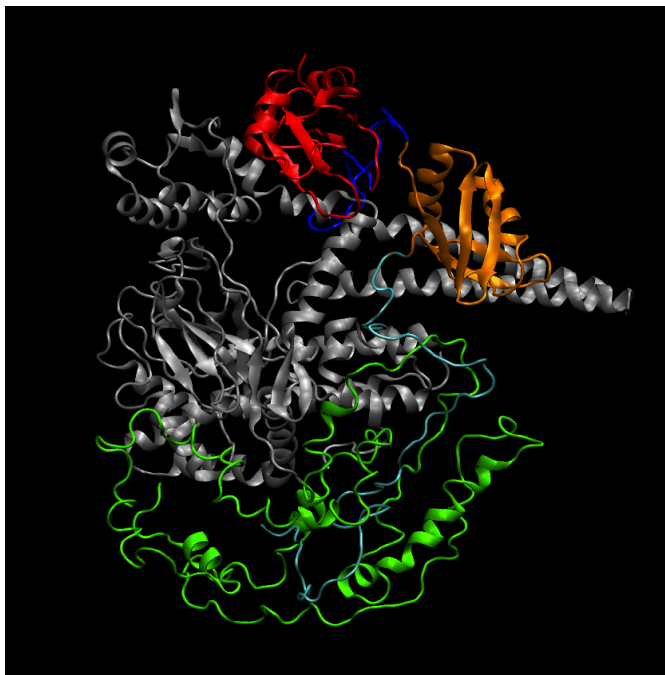
    ShowMessage "This would allow you to look at a protein, down to
the scale of individual atoms..."
AutoMoveToObj gprotein,X=0,Steps=150
Wait (Button)
if Help
    ShowMessage "You could zoom in as closely as you want..."
AutoMoveToObj gprotein,Z=108,Steps=200
Cell 50,50,50
PosObj SimCELL,0,0,108
CleanObj gprotein
# Start Simulation
ForceField Amber96
Cutoff 5.24
TempCtrl Rescale
Temp 25C
Sim On
Wait (Button)
if Help
    ShowMessage "..and watch the atoms as they tumble through space
and time,gaining invaluable knowledge about protein structure and
dynamics."
AutoMoveTo Z=12.8,Steps=600
# MovieSnapshot
# Fly away
Wait (Button)
if Help
    ShowMessage "That is the basic idea behind molecular dynamics
simulations: Use the computer as a microscope to look into a virtual
world."
Sim Off
AutoMoveTo Z=350,Steps=100,Wait=No
Wait 60
AutoRotate Y=3
Wait 60
AutoMoveTo Z=40,Steps=30
AutoRotate Y=-0.1
AutoMove Y=0.1,Z=1
Wait 200
AutoMove Y=2,Z=1
Wait 200

```

**Ćwiczenie 3.** Na podstawie poznanego kodu, studenci zrobią krótką interaktywną 3 minutową prezentację w 3D.

Opis i funkcje programu *Visual Molecular Dynamics* - VMD stworzonego w University of Illinois (<http://www.ks.uiuc.edu/Research/vmd/>).

Jest to program do wizualizacji, animacji i analizy dużych układów biologicznych. Program współpracuje z programem NAMD do MM i MD, dlatego można analizować otrzymane wyniki w programie VMD. Ponieważ studenci wykorzystywali już ten program na wcześniejszych zajęciach (zajęcia z VR, jak również na zajęciach z Modelowania Molekularnego i Obliczeniowej Biologii Strukturalnej na III roku studiów licencjackich specjalność Modelowanie Molekularne i Bioinformatyka), na tym wykładzie zostaną omówione tylko narzędzia do tworzenia animacji jak również dostępne tryby wizualizacji 3D.

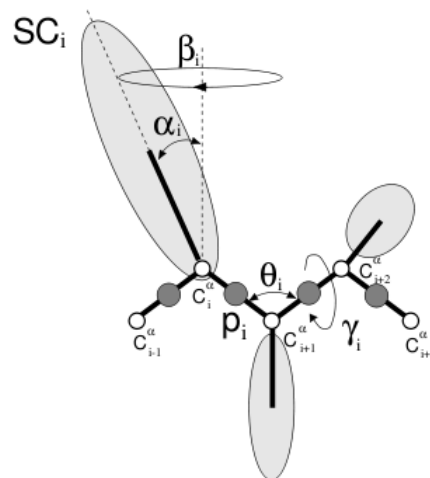


Topoizomeraza I w kompleksie z białkiem ASF. Rysunek jest zrobiony w VMD.

**Ćwiczenie 4.** Na podstawie poznanych narzędzi studenci wygenerują dwa 1-minutowe filmy. Pierwszy stworzą sami dostępnymi narzędziami w VMD, drugi przygotowują z plików pochodzących z programu NAMD (otrzymanych od prowadzącego). Oba filmy będą wyświetlane w 3D. Należy podkreślić, że program NAMD jest znany studentom studiów licencjackich specjalności “Modelowanie Molekularne i Bioinformatyka” Wydziału Fizyki jak i studentom makrokierunku “Bioinformatyka i biologia systemów”.

## VI. Metody modelowania złożonych układów biomolekularnych z wykorzystaniem modeli mezoskopowych, m.in. UNRES

UNRES [1,2], czyli UNited RESidue, jest programem z polem siłowym, zaprojektowanym przez zespół profesorów Adama Liwo z Politechniki Gdańskiej oraz Harolda Scheragę z Cornell University w USA, służącym do gruboziarnistej dynamiki molekularnej białek, w szczególności do wyznaczania ich struktury natywnej. Pole siłowe reprezentuje jeden aminokwas poprzez maksimum dwa centra oddziaływania: środek masy łańcucha bocznego



aminokwasu (SC na rys. obok) oraz środek wiązania peptydowego (p narys. obok). Całkowity potencjał oddziaływania jest dość skomplikowaną funkcją uwzględniającą klasyczne człony obecne w typowych pełnoatomowych polach siłowych (np zawiązane z tzw. potencjałami wiążącymi nałożonymi na wartości kątów płaskich i dwuściennych między wiązaniami (\$U\_b\$ i \$U\_{tor}\$ we wzorze poniżej)) oraz dodatkowe poprawki uwzględniające oddziaływania wielociałowe (człony z indeksami górnymi \$\ge 3\$) [3]. Potencjał ten (wzór poniżej), wyznacza dynamikę atomów węgla \$C\_\alpha\$ (jako geometrycznych centrów oddziaływania, chociaż same atomy \$C\_\alpha\$ słabo oddziałują) oraz środków łańcuchów

aminokwasów.

$$\begin{aligned}
 U = & w_{SC} \sum_{i < j} U_{SC_i SC_j} + w_{SCp} \sum_{i \neq j} U_{SC_i p_j} + \\
 & w_{PP}^{VDW} \sum_{i < j-1} U_{P_i P_j}^{VDW} + w_{PP}^{el} f_2(T) \sum_{i < j-1} U_{P_i P_j}^{el} + \\
 & w_{tor} f_2(T) \sum_i U_{tor}(\gamma_i) + w_{tor} f_3(T) \sum_i U_{tor}(\gamma_i, \gamma_{i+1}) + \\
 & w_b \sum_i U_b(\theta_i) + w_{rot} \sum_i U_{rot}(\alpha_{SC_i}, \beta_{SC_i}) + \\
 & w_{bond} \sum_i U_{bond}(d_i) + w_{corr}^{(3)} f_3(T) U_{corr}^{(3)} + w_{corr}^{(4)} f_4(T) U_{corr}^{(4)} + \\
 & w_{corr}^{(5)} f_5(T) U_{corr}^{(5)} + w_{corr}^{(6)} f_6(T) U_{corr}^{(6)} + w_{turn}^{(3)} f_3(T) U_{turn}^{(3)} + \\
 & w_{turn}^{(4)} f_4(T) U_{turn}^{(4)} + w_{turn}^{(6)} f_6(T) U_{turn}^{(6)}
 \end{aligned}$$

Typy obliczeń, jak i możliwe parametry symulacji opisane są w dokumentacji UNRES [4]. Pomocne są też przykładowe pliki wejściowe (zawierające parametry symulacji) oraz przykładowy skrypt `unres.sh` ustawiający niezbędne zmienne środowiskowe (w szczególności ścieżki do plików z parametrami pola siłowego).

Od strony technicznej, kod UNRESa jest głównie napisany w języku Fortran. Działa w trybie sekwencyjnym lub równoległym, przy czym w symulacjach z wymianą replik zrównoleglenie jest wielostopniowe [5].

Interaktywne połączenie między UNRESem a VMD<sup>1</sup> odbywa się poprzez wirtualny transport danych między otwartymi gniazdami serwera (uruchomiony proces UNRES) i klienta (VMD). Służą do tego funkcje dostępne w pakiecie VMD, zadeklarowane w `vmdsocket.c` i `imd.c`. Fortran bezpośrednio nie obsługuje komunikacji między gniazdami, dlatego do komunikacji między fortranowym kodem UNRESa a kodem C++ VMD do `vmdsocket.c` dodane zostały definicje dwóch funkcji `init_socket` oraz `send_coords` (oraz ich deklaracje w pliku nagłówkowym `vmdsocket.h`). Pierwsza z nich otwiera gniazdo serwera, druga przekazuje aktualne współrzędne atomów na ekran programu klienta (czyli VMD). By komunikacja zadziałała, UNRES powinien być skompilowany ze współdzielonymi funkcjami VMD, czyli z `vmdsocket.c`, `imd.c` oraz odpowiadającymi im plikami nagłówkowymi.

Wywołanie funkcji `init_socket` odbywa się w głównym programie źródłowym UNRESa, czyli `unres.F`. W trybie interaktywnym symulacji, `init_socket` startuje serwer, otwiera gniazdo i czeka na połączenie z klientem (czyli właściwe obliczenia UNRES nie wystartują, dopóki VMD nie nawiąże połączenia z serwerem).

Połączenie klienta z (uruchomionym) serwerem odbywa się już poprzez interfejs VMD<sup>2</sup>, w zakładce *Menu>Extensions>Simulation>IMD Connect*. Zakładka *Simulations* zawiera odnośniki do zestawu okien kontrolnych pozwalających przygotować i uruchomić zadania programu NAMD<sup>3</sup> i połączyć się z uruchomionym serwerem. Do połączenia wystarczy wpisać nazwę lub adres IP komputera-serwera i port, na którym gniazdo nasłuchuje. W przypadku UNRES używany jest port 54321 (obecnie wpisany na sztywno do kodu w `vmdsocket.c`).

Po nawiązaniu połączenia, serwer (UNRES) rozpoczyna obliczenia, wysyłając co pewien czas aktualne współrzędne atomów symulowanej cząsteczki. (Struktura cząsteczki musi być uprzednio wczytana do VMD: *Menu>File>New Molecule*. UNRES może generować struktury m.in. w formacie PDB i to ten format należy wczytać. Czyli strukturę PDB w reprezentacji UNRES należy przygotować wcześniej.). VMD aktualizuje współrzędne na ekranie.

Za wysyłanie współrzędnych z serwera do klienta odpowiedzialna jest funkcja `send_coords`. Wywoływana jest w głównej procedurze odpowiadającej za dynamikę układu, MD, zdefiniowanej w `MD_A-MTS.F`.

---

<sup>1</sup> <http://www.ks.uiuc.edu/Research/vmd/>

<sup>2</sup> <http://www.ks.uiuc.edu/Research/vmd/current/ug/>

<sup>3</sup> <http://www.ks.uiuc.edu/Research/namd/>



## Literatura

1. Liwo, S. Oldziej, M.R. Pincus, R.J. Wawak, S. Rackovsky, H.A. Scheraga. *A united residue force field for off-lattice proteinstructure simulations. I: Functional forms and parameters of longrange sidechain interaction potentials from protein crystal data.* J. Comput. Chem., 18:849873, 1997
2. A. Liwo, M.R. Pincus, R.J. Wawak, S. Rackovsky, S. Oldziej, H.A. Scheraga. *A united residue force field for off-lattice proteinstructure simulations. II: Parameterization of local interactions and determination of the weights of energy terms by Zscore optimization.* J. Comput. Chem., 18:874887, 1997
3. A. Liwo, M. Khalili, C. Czaplewski, S. Kalinowski, S. Ołdziej, K. Wachucik, and Harold A. Scheraga. *Modification and Optimization of the United-Residue (UNRES) Potential Energy Function for Canonical Simulations. I. Temperature Dependence of the Effective Energy Function and Tests of the Optimization Method with Single Training Proteins.* J. Phys. Chem. B 111:260, 2007
4. A. Liwo, S. Ołdziej, C. Czaplewski, D. S. Kleinerman, P. Blood, and Harold A. Scheraga. *Implementation of Molecular Dynamics and Its Extensions with the Coarse-Grained UNRES Force Field on Massively Parallel Systems: Toward Millisecond-Scale Simulations of Protein Structure, Dynamics, and Thermodynamics.* J. Chem. Theory Comput. 6:890, 2010

Dokumentacja program UNRES. Dostępna albo ze strony [www.gro.tri.edu/~liwo](http://www.gro.tri.edu/~liwo) grupy prof. Liwo, albo jako plik unres\_all.pdf w pokatalogu doc/ katalogu instalacyjnego UNRES

## Ćwiczenia

### Zadanie 1.

Okienko IMD Connection pozwala wyświetlać niektóre bieżące dane z symulacji uruchomionej w programie NAMD (np bieżący krok czasowy, energię, jej składowe, temperaturę). Zadanie polega na stworzeniu odpowiednich zmiennych w kodzie UNRES i przekazanie ich wartości do klienta, tak by widoczne były w IMD Connection.

Stworzyć nowe okno/zestaw okien, widocznych w VMD jako *Menu>Extensions>UNRES IMD*, obsługujących połączenie między serwerem (UNRES) a klientem (VMD). Należy do tego wykorzystać bibliotekę Tcl/Tk Pythona. Dokumentacja VMD zawiera też niezbędne wskazówki<sup>4</sup>.

### Zadanie 2.

Interaktywne połączenie między VMD a UNRESem jest obecnie możliwe tylko w trybie sekwencyjnym UNRESa. Należy dodać możliwość połączenia z serwerem UNRES działającym w trybie równoległym.

### Zadanie 3.

Dodać możliwość obsługi zwrotnych komunikatów (VMD → UNRES) (np poprzez funkcję `get_foorces`, która zapewni odbiór przez serwer danych wysyłanych przez klienta), czyli możliwość aktualizacji położeń atomów w oparciu o przyłożone z zewnątrz, przez haptic device, poprzez interfejs VMD, siły.

---

<sup>4</sup> <http://www.ks.uiuc.edu/Research/vmd/current/ug/node115.html>

## VII. Mathematica i techniki VR

Mathematica (<http://www.wolfram.com/>) obsługuje urządzenia haptyczne o których była mowa w poprzednich paragrafach. W szczególności rozwijany jest projekt interaktywnej pracy ze strukturami molekularnymi, patrz:

<http://www.wolfram.com/technology/guide/BuiltInGamePadHIDSupport/UseTheLatestHapticDevices.html>

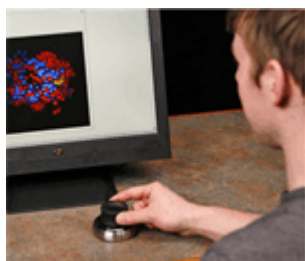


Mathematica współpracuje też z innymi urządzeniami HID (*Human Interface Devices*) <http://www.wolfram.com/mathematica/features/interactivity-and-interface-design.html>.

Pozwala to na interakcyjną pracę obiektami graficznymi 3D. Rozpoznawanie i konfiguracja gamepad'ów, joystick'ów, urządzeń haptycznych oraz myszy 3D odbywa się automatycznie.

Więcej informacji, patrz:

<http://www.wolfram.com/technology/guide/BuiltInGamePadHIDSupport/>



Obracanie grafiki 3D z wykorzystaniem 6-ciu stopni swobody.



Automatyczna kontrola z wykorzystaniem dynamicznych interface'ów.



Współpracujące gamepady i joysticki

Na uwagę zasługuje fakt, że *m2wrl.m* (*Mathematica to wrl VRML format*) dostarcza zbioru funkcji do konwersji obiektów 3D do języka wirtualnego modelowania (*Virtual Reality Modeling Language* (VRML)), patrz: <http://www.lrdev.com/lr/math/m2wrl.html> .

## *Ćwiczenia.*

### *Zadanie 1*

Ze stron Wolfram Research (<http://www.wolfram.com/>) ściągnąć na stację roboczą przykłady dydaktyczne struktur molekularnych. Zrealizować operacje zalecane w opisach do tych struktur.

### *Zadanie 2*

Z serwisu Protein Data Bank (<http://www.rcsb.org/>) załadować struktury kilku niewielkich białek. Wczytać współrzędne w środowisku Matematyki i wygenerować ich obrazy 3D.