

WARSZTAT: SOLID EDGE

Firma Intergraph jest dobrze znana w świecie ze swoich produktów i nie trzeba jej nikomu specjalnie przedstawiać – obok innych znanych producentów to jeden z liderów w świecie grafiki komputerowej i twórców oprogramowania CAD na potrzeby szerokiej rzeszy inżynierów przeróżnych specjalności (Solid Edge został sprzedany firmie Unigraphics Solutions, która właśnie wprowadziła na rynek siódmą wersję programu – przyp. red.).

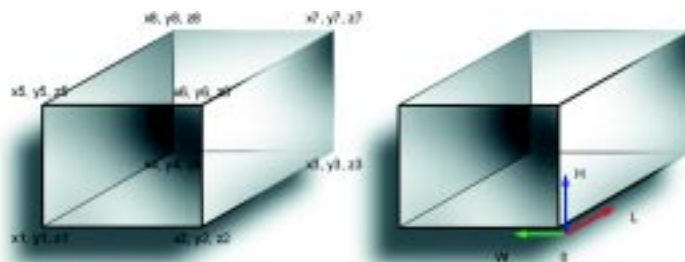
Ponieważ tak się ostatnio złożyło, iż musiałem wyciągnąć z archiwum (czyt. szafy) i odkurzyć mój stary projekt jednostopniowej przekładni zębatej, który wykonywałem kiedyś na zajęciach jeszcze w czasie studiów, nadarzyła się dobra okazja, aby przypomnieć sobie, z jakimi zagadnieniami zetknąłem się w czasie tworzenia owego projektu. Postanowiłem jednak, że tym razem odtworzę go oczywiście – jak przystało na projektanta dzisiejszych czasów – nie na desce kreślarskiej, lecz przy pomocy niezastąpionego w takiej sytuacji komputera. Pozostał jedynie problem wyboru narzędzia, lecz po dłuższym namyśle zdecydowałem się ostatecznie na Solid Edge. Program i moje wrażenia po pierwszym z nim kontakcie chciałbym przedstawić na łamach niniejszego artykułu. Przyznam, że na program zwróciłem uwagę zarówno pod wpływem bardzo pochlebnych opinii znajomych, którzy gorąco zachęcali mnie do zapoznania się z tym interesującym pakietem, jak również zwykłej chęci spotkania się z czymś nowym. Prawdę mówiąc – kiedy po raz pierwszy uruchomiłem program, byłem nieco... zdezorientowany. Na pierwszy rzut oka brak tu było wielu niezbędnych funkcji i narzędzi, do których od dawna przyzwyczajali mnie producenci programów CAD. Nieco odmienny interfejs, skromna liczba ikon na pulpicie również nie wzbudzały mojego zachwytu (przyzwyczajenia to naprawdę druga natura człowieka!). W tym miejscu wypada jednak zaznaczyć, iż nie miałem do czynienia z najnowszą wersją programu. Tak czy owak, zastanawiałem się, czy w tym środowisku, zorientowanym zgodnie z zapewnieniami na maksymalizację wydajności, można rzeczywiście pracować efektywnie. Ale nie uprzedzajmy faktów, tak więc... kurtyna w górę!

Solid Edge (podobnie jak i wiele innych tworzonych dziś aplikacji) jest programem opartym na technologii parametrycznego modelowania bryłowego. Jedną z podstawowych korzyści wynikających z tego faktu, w porównaniu do modelowania powierzchniowego i krawędziowego, jest szybkość, z jaką koncepcyjne modele mogą być generowane. Dodatkowo historia tego, w jaki sposób tworzymy nasz model, przechowywana jest w pliku projektowym. Tak więc nasz projekt posiada dostateczne zasoby inteligencji, aby odtworzyć model bryły w przypadku, gdy:

- ⊕ konieczne są zmiany konstrukcyjne;
- ⊕ wymagana jest analiza wytrzymałościowa, np. metodą elementu skończonego (MES);
- ⊕ stosowane są obrabiarki sterowane numerycznie;
- ⊕ wymagane jest wykonanie prototypu stereolitograficznego;
- ⊕ wymagana jest kontrola konfliktów między poszczególnymi częściami konstrukcji.

We wzajemnym powiązaniu bryły parametryczne i ich właściwości mogą być użyte do zautomatyzowania procesu projektowego, bowiem od momentu, w którym zostały utworzone, otrzymaliśmy model bryłowy, bazujący na intencjach projektowych konstruktora, a nie po prostu zbiór punktów umieszczonych w przestrzeni. W odróżnieniu od modelu krawędziowego czy powierzchniowego, które są po prostu jedynie obrazami konstrukcji, model bryłowy stanowi już sam w sobie komputerową jej symulację. W modelu bryłowym parametryczny model budowany jest w oparciu o parametry, które służą do jego tworzenia, a nie w oparciu o punkty definiujące położenie jego wierzchołków. Tak więc np.

zamiast żmudnego definiowania krawędzi i powierzchni możemy wydać polecenie: wywierć otwór w danym miejscu, o takiej głębokości (lub na wylot) i kształcie zgodnym z taką lub inną normą. Korzyścią z tego faktu jest również pewne przybliżenie do rzeczywistego procesu technologicznego w czasie tworzenia przedmiotu. Różnice w podejściu do tematu ilustruje rys. 1.



Rys. 1. Reprezentacja sześcianu w modelu powierzchniowym (z lewej) i w modelu bryłowym (z prawej), gdzie O – początek, L – długość, H – wysokość, W – szerokość bryły.

Bryła może być edytowana poprzez modyfikację jej zdefiniowanych parametrów, podczas gdy jedynym sposobem edycji modelu powierzchniowego jest przemieszczanie punktów. Właściwości i bryłowy opis są bardziej inteligentne niż geometria powierzchni i wierzchołków – model bryłowy jest jednoznaczniejszą reprezentacją. Oto kilka przykładów korzyści wynikających z modelowania bryłowego:

- ⊕ Otwór zdefiniowany jako przelotowy nadal będzie przechodził na wskroś przez bryłę, nawet jeśli grubość bryły zostanie zmieniona.
- ⊕ Gdy manipulujemy sześcianem utworzonym narzędziami modelowania powierzchni, brak jest informacji określającej, czy jest on pusty (wydrązo-

ny), czy też jest bryłą pełną (przy modelowaniu powierzchniowym w niektórych programach mamy możliwość wprowadzenia takiej informacji ręcznie). W modelowaniu bryłowym sprawa jest oczywista. Ponadto, jak już wspomniałem wcześniej, ze względu na sposób reprezentacji nie mamy możliwości bardziej inteligentnej modyfikacji obiektu;

- ✦ Jeśli okaże się, że umieściliśmy otwór nie tam, gdzie powinien się znajdować, wystarczy tylko zmienić zadane wartości początkowe (cechy otworu) i program automatycznie skoryguje kształt elementu, wyznaczy nowe przekroje itd.

Na zakończenie tego punktu może jeszcze tylko kilka uwag. Ponieważ model bryłowy jest komputerową symulacją, nie możemy dopuszczać do powstawania błędów w procesie modelowania bryły. W modelowaniu powierzchniowym model może mieć dwie krawędzie, które nie łączą się dokładnie. Gdy nieciągłość stwierdzana jest jedynie przy dużym powiększeniu, może być ona dopuszczalna, ponieważ błąd nie jest widoczny dla gołego oka. Natomiast w modelowaniu bryłowym ta sama nieciągłość spowoduje wystąpienie błędów. Błędy, początkowo nie zauważone przez nas, powodują odmowę wykonania pewnych operacji w dalszych etapach projektowania. Parokrotnie miałem okazję się o tym przekonać, kiedy program, wpędzając mnie tym samym w głęboką frustrację, uparcie nie chciał wykonać prostego fazowania czy zaokrąglania krawędzi. Musimy więc uważać, aby zadawane parametry pokrywały się z rzeczywistością i np. promień zaokrąglenia nie przekraczał wielkości jednej z krawędzi. Innym skutkiem ubocznym modelowania parametrycznego w porównaniu do metod tradycyjnych jest wzrost zapotrzebowania na moc obliczeniową i zasoby pamięciowe komputera, co wyraźnie uwidoczniło się, kiedy próbowałem zmieniać parametry wcześniejszych operacji – zwłaszcza tych z początkowej fazy projektowania. W miarę postępu prac czas potrzebny na przemyślenie zmian przez procesor wydłużał się znacznie.

SOLID EDGE – PART

Środowisko pracy podzielone zostało na trzy zasadnicze części – PART (budowa elementu), ASSEMBLY (budowa zespołów z wcześniej przygotowanych elementów) oraz DRAFT (tworzenie rysunków do dokumentacji technicznej). Zgodnie z logiką postępowania zaczynamy od pierwszej części – PART – przeznaczonej do generowania podstawowych elementów składowych naszego mechanizmu, choć możliwe jest tworzenie części i generowanie krawędzi w kontekście pracy w środowisku ASSEMBLY. Kiedy np. rysujemy profil przy aktywnej opcji *In-place*, możemy dołączyć krawędzie z innych aktywnych lub nieaktywnych części. Kiedy z kolei wyznaczamy profil otworu, możemy budować jego fragmenty, używając krawędzi innych elementów. Mechanizm ACIS 2.0, w oparciu o który funkcjonuje środowisko, wprowadza istotne – z punktu widzenia projektanta – cechy, takie jak zaokrąglania, fazowania krawędzi czy generowanie cienkich ścianek, włączając w to przypadki, kiedy wzajemnie się na siebie nakładają. Dzięki ACIS 2.0 dodatkowo zwiększono dokładność generowanych przez program parametrów elementu przy jednoczesnym

zmniejszeniu rozmiarów pliku wyjściowego. Należy tu przyznać, że mechanizm ten działał prawie bez zarzutu i podczas moich prób program komunikował o błędach kiedy rzeczywiście zadawane przeze mnie parametry odbiegały nieco (lub nawet więcej) od zasad logiki – jak wspomniałem wcześniej, to my jesteśmy odpowiedzialni za to aby projektowany przedmiot nie odbiegał konstrukcyjnie od realiów.

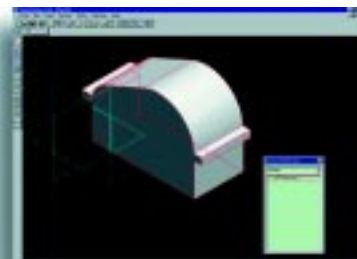
TWORZENIE ELEMENTÓW

Ze względów oczywistych pominiemy zagadnienia związane z wytrzymałościowymi obliczeniami wstępnymi – nie leży to w tematyce artykułu – pamiętajmy jednak, że to one decydują o kształcie i gabarytach projektowanych części. Zaczniemy zatem budowę przekładni od elementu najbardziej chyba skomplikowanego – obudowy. Użyjmy ikonki *Protrusion* (wyciągnięcie). Po wybraniu odpowiedniego planu roboczego program automatycznie przeniesie nas (tam i z powrotem) w tryb pracy *Profile* (profil), co jest zresztą bardzo wygodnym i efektywnym rozwiązaniem. W oparciu o aktywny plan wygenerujemy przekrój jak na rysunku 2.



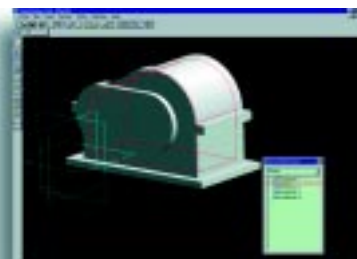
Rys. 2. Edycja kształtu profilu wyjściowego.

Bardzo przyjazny dla użytkownika okazał się interfejs przy projektowaniu kształtu przekroju. Automatycznie pojawiające się linie odniesienia do innych elementów, dociąganie do ich końców, stycznych czy punktów centralnych okręgów znacznie poprawiają wydajność i komfort pracy – momentami musiałem wyłączać niektóre opcje, gdyż robiło się zbyt gęsto na ekranie. Dodatkowo sporą część informacji o tym, w jakim znajdujemy się stadium operacji czy położeniu, pokazuje zmieniający się dynamicznie kursor. W zależności od środowiska, w którym się znajdujemy – automatycznie pojawiają się odpowiednie dla danego kontekstu paski narzędziowe. Nie zaśmiecamy więc cennego miejsca na ekranie niepotrzebnymi narzędziami – program wie, czego w danej chwili moglibyśmy potrzebować. Mając zdefiniowany przekrój, wystarczy kliknąć *Finish* i jeśli w czasie edycji nie popełniliśmy błędów, to jesteśmy wyrzuceni z powrotem do środowiska 3D. Tutaj wystarczy podać tylko długość wyciągnięcia (dynamicznie kursorem lub z klawiatury) i otrzymamy następujący obiekt (rys. 3).



Rys. 3. Efekt użycia komendy PROTRUSION.

Wewnątrz zrobimy wycięcie dla uzyskania odpowiedniej grubości ścianek – metoda *Cutout*, następnie dodajmy podstawę i symetryczne naddatki na gniazda łożysk (*Protrusion*, rys. 4).



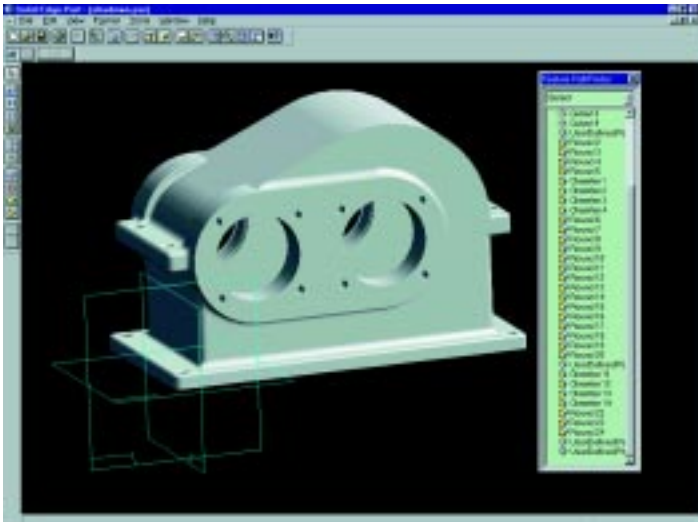
Rys. 4.



Rys. 5

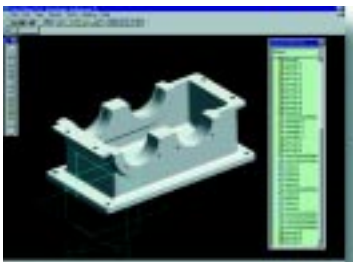
Z grubsza kształt obudowy jest już zdefiniowany, jeszcze tylko kilka otworów – *Hole* (rys. 5) i ostateczne wykończenie – *Hole* (otwór), *Chamfer* (fazowanie krawędzi), *Round* (zaokrąglenie krawędzi) oraz *Revolved Cutout* (obrotowe wycięcie profilu) – rysunek 6. Przy okazji definiowa-

nia zaokrągleń okazało się, jak istotna jest kolejność i zakres poszczególnych operacji. Od tych dwóch czynników oraz od typu wybranych zaokrągleń w dużej mierze zależało czy zakończy się ona sukcesem, czy komunikatem o błędzie.



Rys. 6

Teraz nie pozostało już w zasadzie nic innego, jak tylko przekroić naszą obudowę na dwie części... i tu niespodziewanie pojawił się problem. W żaden sposób nie mogłem tego dokonać przy pomocy jednej operacji i musiałem się solidnie napocić, aby uzyskać efekt końcowy – czyli górną pokrywę i podstawę korpusu. Nie byłem pewien, czy nie popełniłem błędu strategicznego na samym początku, rozpoczynając projektowanie obudowy jako jednolitego elementu. W ostatecznym jednak rozrachunku udało się dzięki temu zaoszczędzić sporo czasu, nie tracąc go na ponowne definiowanie powtarzających się dla obu części operacji. Ok... jeśli nie tak, to inaczej. Pożądany efekt przecięcia uzyskałem, kasując po kawałku najpierw jedną, a potem drugą połowę, poprzez kilkakrotne użycie *Cutout* (program odmawiał wykasowania całej połówki za jednym pociągnięciem). Nie była to niestety metoda zbyt elegancka, choć mogła być konsekwencją popełnionego gdzieś wcześniej przeze mnie błędu. Końcowy efekt widać na rysunkach 7 i 8.



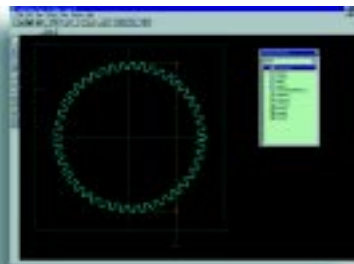
Rys. 7. Podstawa korpusu.



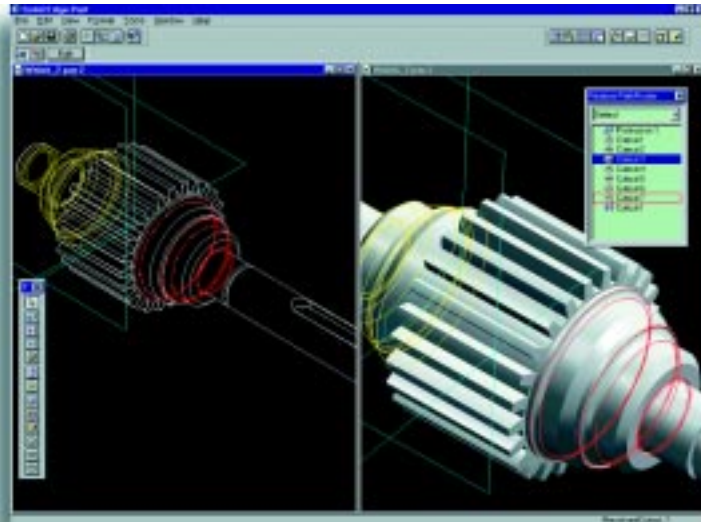
Rys. 8. Pokrywa.

Stworzymy teraz inne elementy podzespołu: wałek uzębiony, koło zębate i łożysko (wszystkich detali na potrzeby przykładu nie ma sensu generować). Wałek uzębiony wykonamy w trzech operacjach! Zaczniemy od zdefiniowania profilu zębów dla komendy wyciągnięcia – to będzie najtrudniejsza część operacji... (rys. 9)

Następnie zdefiniujemy odpowiedni profil, zastosujemy dwukrotnie obrotowe wycięcie i... w zasadzie to już koniec! Efekt jak poniżej – rysunek 10.

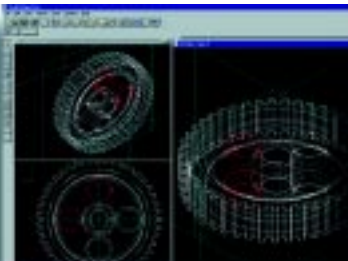


Rys. 9. Profil wyjściowy dla wałka uzębionego.



Rys. 10. Dzięki modelowaniu parametrycznemu uzyskaliśmy powyższy kształt w trzech operacjach.

Nieco więcej operacji wymaga stworzenie koła zębatego. Zaczniemy, podobnie jak w przypadku poprzedniego elementu, od zdefiniowania profilu zębów, następnie operacji wyciągnięcia. Otrzymaliśmy zarys płaskiego, pełnego, uzębionego walca. Wokół osi symetrii dokonujemy obustronnych wycięć, centralnie i po obwodzie wiercimy otwory, zaokrąglamy wewnętrzne krawędzie, do centralnego otworu dodajemy wycięcie z myślą o osadzeniu koła na wpuszcisku – rys. 11 i 12.

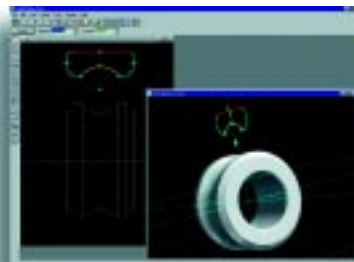


Rys. 11. Operacje przy tworzeniu koła zębatego.

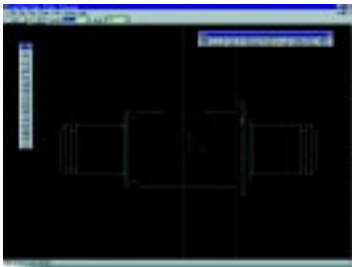


Rys. 12.

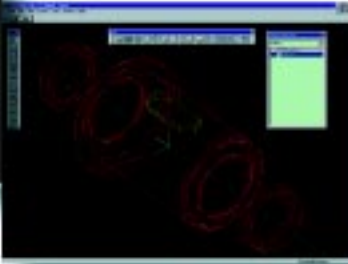
Łożysko również nie przedstawia trudności – dwa obrotowe profile tworzące wewnętrzny i zewnętrzny pierścień (w uproszczeniu) oraz dodatkowo, również obrotowy profil definiujący połowę przekroju jednej kulki – pozostałe utworzymy później, korzystając z polecenia *Circular Pattern* (rys. 13).



Rys. 13.



Rys. 14.



Rys. 15.

Utworzenie wałka pod koło zębate jest wręcz banalne – odpowiednio generujemy obrotowy przekrój i jedno wycięcie. (rys. 14 i 15).

Warto nadmienić, że dla zdefiniowanego elementu możemy przeprowadzić analizę cech fizycznych (rys. 16).



Rys. 16

Po określeniu gęstości materiału, z którego ma być wykonany element, program oblicza masę, objętość, pole powierzchni, a także wyznacza środek ciężkości, momenty bezwładności względem poszczególnych osi układu. Jest to bardzo przydatna cecha, biorąc pod uwagę możliwość wykorzystania naszego modelu w dalszej analizie czy korekcie pierwotnych założeń konstrukcyjnych.

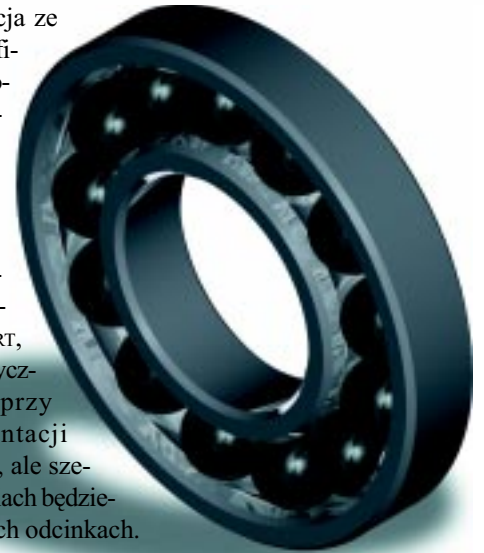
WYMIANA DANYCH Z INNYMI APLIKACJAMI

O jakości aplikacji często decyduje to, czy potrafi porozumieć się z innymi programami. Osobiście lubię mieć możliwość rozpoczęcia pracy w jednym, a skończenia w innym programie. Przyznam, że Solid Edge pod tym względem mnie nie rozczarował. Poprzez zapis pracy w formacie SAT bez problemów udało mi się wysłać wygenerowane pliki do Mechanical Desktop, gdzie można kontynuować pracę lub np. opracować bardziej realistyczny rendering elementu (rys. 17 i 18). Inną cechą jest dale-



Rys. 17.

ko posunięta integracja ze środowiskiem MS Office. Umożliwia to odwoływanie się do baz danych np. przy projektowaniu cech elementu. Mało tego, zmiany w bazie danych, powodują zmianę konstrukcyjną części w środowisku PART, co z kolei jest automatycznie uwzględniane przy tworzeniu dokumentacji w środowisku DRAFT, ale szerzej o tych możliwościach będziemy pisać w następnych odcinkach.



Rys. 18.

KILKA UWAG

Przez cały czas tworzenia projektu towarzyszył mi nieodzownie *Feature Pathfinder* (gdzie ja tę nazwę słyszałem?). W okienku tym zapisywana jest cała historia dokonywanych zmian i operacji. Mimo iż charakterystyczne cechy obiektu podświetlane są dynamicznie, jeśli przemieszczamy nad nimi kursor, to często podczas edycji obiektu trudno jest zaznaczyć tę właściwą, interesującą nas (zwłaszcza jeśli element jest już bardzo skomplikowany). Tu właśnie *Pathfinder* okazał się bardzo pomocny – wystarczy zaznaczyć kursorem odpowiednią nazwę operacji i możemy przejść do jej edycji. Po nabraniu nieco wprawy traktowałem długość tej listy jako swojego rodzaju wskaźnik optymalizacji ilości wykonywanych zadań – im krócej, tym lepiej.

Z kolei specjalne okno dialogowe *Properties* umożliwia ustawienie i zapisanie wielu dodatkowych informacji o wykonywanym projekcie, ustalenie jednostek miar, statusu elementu w odniesieniu do całego projektu (rys. 19).

To tyle jeśli chodzi o tematykę pierwszej części artykułu, w kolejnym odcinku zajmiemy się tworzeniem całościowych zespołów z przygotowanych wcześniej elementów, a następnie generowaniem dokumentacji technicznej. Przepraszam jednocześnie, że nie facytowałem się tłumaczeniem na język polski wszystkich obcych nazw, co w świetle ustawy o ochronie języka jest dalece niepożądane – ale niech tam... w końcu piszemy dla fachowców!



Rys. 19.

Rajmund Michałowski
mundek@3d.pl