Jak wybrać akcelerator 3D dedykowany aplikacjom typu MCAD?

Clive „Max" Maxfield

*Dla aplikacji typu MCAD wydajność karty graficznej jest czynnikiem krytycznym. W artykule tym znajdziecie opis cech akceleratorów 3D, które będą pomocne przy zakupie karty graficznej stosownej do waszych aplikacji typu MCAD (**Mechanical Computer Aided Engineering**).*

Sprawa wyboru specjalizowanych kart graficznych do zastosowań przy skomplikowanej wizualizacji obiektów 3D i dedykowanych takim dziedzinom jak komputerowe wspomaganie prac inżynierskich (***CAD***) jest często traktowana niewłaściwie. Wynika to zapewne z braku informacji o danym produkcie jak również ze skąpej oferty dystrybutorów. Postanowiliśmy przybliżyć zainteresowanym czytelnikom niezbędne prawidła jakimi powinni kierować się przy wyborze high-end'owych akceleratorów 3D.
Przed dokonaniem zakupu karty graficznej jak i całego systemu użytkownik powinien jasno i wyraźnie określić wymagania dotyczące indywidualnych warunków pracy, w skład których należy zaliczyć, profil działalności, środowisko programowe, system operacyjny, wymagania sprzętowe, które odnoszą się do podsystemu graficznego.
Tabela 2 ukazuje niektóre bardzo istotne cechy podsystemów graficznych z gałęzi ***high-end*** (specjalizowane oraz specjalistyczne podsystemy graficzne) wymagane w aplikacjach typu ***MCAD*** (***Mechanical Computer Aided Engineering***). Projektanci oraz osoby pełniące stanowiska specjalistów z dziedziny zastosowań programów typu ***CAD/CAM/CAE*** powinni zrozumieć te prawidła, by dokonywane zakupy sprzętu były jak najbardziej właściwe. Artykuł przedstawia w sposób ogólny cechy wymagane przy modelowaniu w przestrzeni 3D i trzeba mieć świadomość, że każda aplikacja MCAD ma swoje własne wymagania dotyczące kart graficznych.

ROZDZIELCZOŚĆ I „GŁĘBIA KOLORU"

W nomenklaturze grafiki komputerowej i podsystemów graficznych, rozdzielczość oznacza ilość pixeli (elementy składowe obrazu) używanych do wyświetlenia bieżącego obrazu. Rozdzielczość obrazu nie odnosi się bezpośrednio do rozmiaru ekranu. Na przykład, jeżeli pracujemy z 1mm pixelem to obraz składający się z 100 x 100 pixeli będzie miał wymiar 10cm x 10cm. Jeżeli jednak użyliśmy $1m^2$ dla naszego pixela to ten sam obraz będzie miał wymiar 100 metrów x 100 metrów. Taki sam obraz możemy wyświetlić na 21'' monitorze jak również i na 17 calowym monitorze z tą sama rozdzielczością – powiedzmy 1024x768 pixeli. Pomimo tego, że obraz wyświetlany na monitorze 21 calowym będzie fizycznie zajmował więcej miejsca, na obydwu monitorach będzie on maił dokładnie taką samą ilość widocznych detali.

(a) 1 - bitowy obraz (b) 8 - bitowy obraz (c) 24 - bitowy obraz

***Rys. 1** Jakość wyświetlanego obrazu jest funkcją ilości bitów użytych do opisu pojedynczego piksela.*

Pojęcie „głębia koloru" odnosi się do ilości bitów użytych do wyświetlenia danego koloru pojedynczego pixela składającego się na całość wyświetlanego obrazu. Jeżeli wykorzystujemy tylko pojedynczy bit tak by reprezentował pojedynczy pixel, wtedy powiedzmy najogólniej bit jest określony tylko dwoma stanami ON lub OFF (logiczne 0 lub logiczna 1). Taka „głębia koloru" pozwala na wyświetlenie tylko dwóch kolorów,

powiedzmy czarny lub biały. Łącząc więcej bitów z pojedynczym pixelem pozwala na wyświetlenie obrazu z szersza paletą barw czyli obraz jest przedstawiony za pomocą większej ilości kolorów. Dwa bity informacji o kolorze pozwalają stworzyć 2^2=4 logicznych zer i jedynek (00, 01, 10, i 11) co jest reprezentowane przez wyświetlenie 4 kolorów na monitorze. Podobnie 8 bitów odwzorowuje 2^8=256 kolorów i tak dalej.
Istnieje kilka najpopularniejszych schematów „głębi kolorów", najlepszy schemat 24 bitowa „głębia koloru" wymaga 24 bitów informacji opisującej pojedynczy pixel: 8 bitów na każdy pixel kolorów podstawowych, czerwony, zielony, niebieski (***RGB***). Schemat ten jest popularnie nazwany „24-bit true color" i pozwala na uzyskanie 16,7 miliona kolorów, w zupełności wystarczających do wyświetlania naturalnych barw.
Naturalnie wraz ze zwiększeniem „głębi koloru" następuje zwiększenie zapotrzebowania na pamięć tzw. „bufora ramki" (***frame buffer***), to samo dotyczy uzyskiwania wyższych rozdzielczości (patrz Tabela 2). Wartości podane w Tabeli 2 oznaczają obliczone wielkości wymaganej pamięci dla „bufora ramki" podaną w MB, będącej rezultatem kombinacji najpopularniejszych schematów głębi kolorów i rozdzielczości ekranu.
Karty graficzne są zwykle wyposażone w 1, 2, 4, 6, 8, 12, 16, lub 32 MB „bufora ramki". Wartości podane w nawiasach w **Tabeli 1** wskazują na niezbędne minimum fizycznej pamięci „bufora ramki".

	Głębia koloru			
Rozdzielczość	8 - bit	16 - bit	24 - bit	32 – bit
640 x 480	0.29 MB (1)	0.59 MB (1)	0.88 MB (1)	1.17 MB (2)
800 x 600	0.46 MB (1)	0.92 MB (1)	1.37 MB (2)	1.83 MB (2)
1024 x 768	0.75 MB (1)	1.50 MB (2)	2.25 MB (4)	3.00 MB (4)
1280 x 1024	1.25 MB (2)	2.50 MB (4)	3.75 MB (4)	5.00 MB (6)
1600 x 1200	1.83 MB (2)	3.66 MB (4)	5.49 MB (6)	7.32 MB (8)
1824 x 1368	1.98 MB (2)	4.76 MB (6)	7.14 MB (8)	9.52 MB (12)
1920 x 1080	2.38 MB (4)	3.96 MB (4)	5.93 MB (6)	7.91 MB (8)

Tab. 1 *Zwiększając rozdzielczość i głębię koloru zwiększa się zapotrzebowanie na pamięć bufora ramki.*
Uwaga: 32 – bitowa głębia koloru ujęta w ostatniej kolumnie jest de'facto kombinacją koloru 24 – bitowego oraz 8 – bitów opisujących przeźroczystość.
Z wyjątkiem medycznych specjalistycznych systemów wyświetlania rozdzielczość 1824 x 1368 jest wyświetlana przez kartę graficzną RealiZm II OpenGL. Rozdzielczość 1920 x 1080 jest dedykowana dla monitorów o formacie wyświetlania 16:9.

W akceleratorach graficznych 3D oferujących zaawansowane funkcje wyświetlania takie jak:
- ***double buffering***
- ***Z – buffering***

„bufor ramki" musi być odpowiednio większy. W świecie idealnym, wszystkie podsystemy graficzne powinny mieć możliwość wyświetlania 24 bitowej „głębi koloru" w każdej żądanej rozdzielczości. Niestety wielu producentów zaniża możliwości wyświetlania obrazu ograniczając „głębię koloru" do 16, 15 lub nawet 8 bitów. W takich przypadkach karat graficzna po prostu odrzuca dodatkową informacje o kolorze, lub wykonuje proces znanym jako „***dithering***" (granulacja) usiłując tylko podnieść jakość wynikowego obrazu na ekranie.
Tworzenie obrazów o fotorealistycznej jakości lub dużych złożeń (skomplikowana geometria) implikuje stosowanie 24 bitowej „głębi kolorów" oraz wysokich rozdzielczości ekranowych po to by wyraźnie opisywać modele zawierające dużą ilość małych szczegółów (np. otwory, fazowania, zaokrąglenia). Niestety pomimo dużego wyboru podsystemów graficznych wiele z nich domaga się wręcz możliwości wyświetlania 24 bitowej „głębi koloru" osiąganej w najwyższych rozdzielczościach i jak wykazuje praktyka często okazuje się iż np. aby uzyskać najwyższą „głębię koloru" należy zrezygnować z wysokich rozdzielczości i na odwrót.

TRANSLUCENCY (Alpha Blending)

Przeźroczystość (***transparency***) jest to taka cecha materiału, obiektu, która pozwala promieniowi świetlnemu (światłu) przeniknąć przez niego, tak iż obiekty znajdujące się za wspomnianym obiektem będą widoczne ostro i wyraźnie.

Półprzeźroczystość (***translucency***) jest natomiast taką cechą materiału która pozwala promieniowi świetlnemu (światłu) przeniknąć i częściowo zostać w nim rozproszona tak iż obiekty znajdujące się za wspomnianym obiektem będą widoczne ale będą wyglądały jak przez mgłę. Właśnie użytkownicy aplikacji typu ***MCAD*** najczęściej używają półprzeźroczystości do celów weryfikacyjnych i prezentacyjnych gdy np. nadaje się zewnętrznym powłokom cechę półprzeźroczystości by móc obserwować niewidoczne fragmenty złożeń.

Aby móc opisać efekt półprzeźroczystości, „bufor ramki" w podsystemach graficznych „***high-end***" dla pojedynczego pixela przechowuje dane (bity informacji) zwane „***alpha bits***".

Podobnie jak przez ilość bitów stosowanych w opisie koloru opisywany jest efekt półprzeźroczystości gdzie np. 2^8=256 poziomów półprzeźroczystości pozwalających osiągnąć efekt zupełnej przeźroczystości (transparency) lub obiektu o przeźroczystości równej 0 (patrz rysunek poniżej).

a) obiekt bez składowej alpha b) obiekt półprzeźroczysty

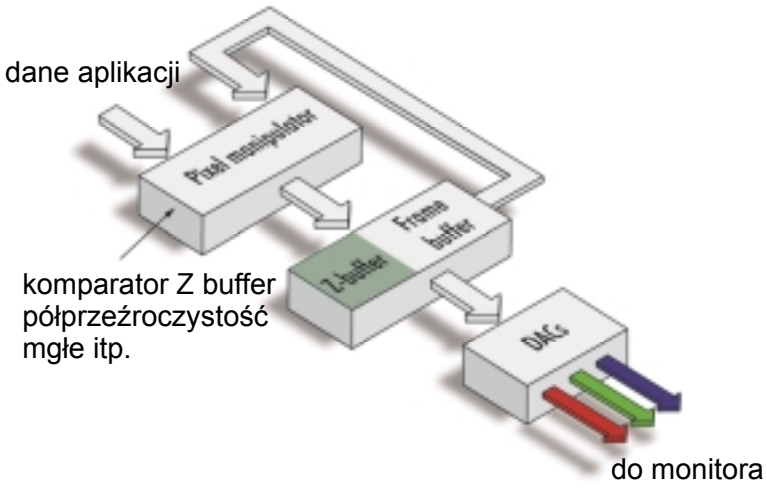
Rys. 2 *Bufor ramki w high-end'owych kartach graficznych używa bitów alpha do wyświetlenia przeźroczystości każdego piksela*

Dlatego w kartach graficznych segmentu high-end „bufor ramki" jest zawsze reprezentowany 32 bitową „głębią koloru" opisującej pojedynczy pixel, po 8 bitów dla każdego koloru podstawowego: czerwony, zielony, niebieski (***RGB*** oraz 8 bitami przeznaczonymi na efekt połprzeźroczysości (***translucency***). Taki schemat głębi kolorów jest często zwany kolor 32 bitowy ***RGBA***.

Z-BUFFERING

Podczas ***renderingu*** sceny 3D komputer przeprowadza matematyczną konwersję obiektów 3D na właściwy dla ekranu monitora widok 2D. W rezultacie pojedynczy pixel posiada trzy współrzędne „XYZ" gdzie współrzędna „Z" określa związaną z oryginalnym modelem 3D „głębię koloru" pixela z punktu widzenia obserwatora.

W tym samym czasie kiedy bity informacji ***RGBA*** skojarzone z pojedynczym pixelem są zapisywane w pamięci (***frame buffer***), wartości współrzędnej „***Z***" pojedynczego pixela zapisywana jest w specjalnym segmencie pamięci „bufora ramki" zwanym „***Z – buffer***". W praktyce zdarza się sytuacja, iż niektóre obiekty w przestrzeni 3D mogą znajdować się przed innymi, wtedy system ma możliwość wielokrotnego przypisania bitów pobieranych z „***Z – buffer***" właściwym pixelom w pamięci – buforze ramki. Jednak aby taka sytuacja mogła zaistnieć wartość współrzędnej „***Z***" przechowywanej w segmencie pamięci „***Z-buffer***" związana z przechowywanym pixelem w buforze ramki (***RGBA***) jest porównywana z wartością współrzędnej „***Z***" nowego pixela. Jeśli nowy pixel znajduje się dalej od obserwatora niż przechowywany pixel, wtedy wartość opisująca nowy pixel jest odrzucana. Jeżeli jednak nowy pixel znajduje się bliżej obserwatora niż przechowywany pixel wtedy bity informacji skojarzone z nowym pixelem „nadpisują" w buforze ramki stary pixel. Opisywaną sytuacje opisuje Rysunek 3.



Rys. 3 *Z – bufor jest częścią bufora ramki.*
Uwaga: Schemat ten jest jedynie schematem poglądowym i nie odnosi się do żadnych konkretnych rozwiązań

Różne podsystemy graficzne użytkują różne ilości bitów do przechowywania informacji składowej „***Z***". Większe obiekty 3D takie jak na przykład samoloty wymagają „***Z-buffora***" o większej głębi (czytaj większej ilości bitów informacji dla pojedynczego pixela). Jeżeli do opisania obiektu 3D została użyta zbyt mała ilość bitów informacji mogą mieć miejsce nieprzewidziane anomalie takie jak wyświetlane obiekty sprawiają wrażenie jakby się nawzajem przenikały. Może mieć również miejsce gorsza sytuacja gdzie wyświetlane obiekty na przemian migotają podczas manipulowania nimi w przestrzeni 3D.
Z praktycznego punktu widzenia minimalna „głębia ***Z-buffer***" instalowana w kartach 3D należących do segmentu kart ***low-end*** wynosi 16 bitów przypadających na pojedynczy pixel podczas gdy w kartach specjalizowanych i ***high-end*** wymagane są zawsze 24 bity lub więcej przypadające na każdy pixel.

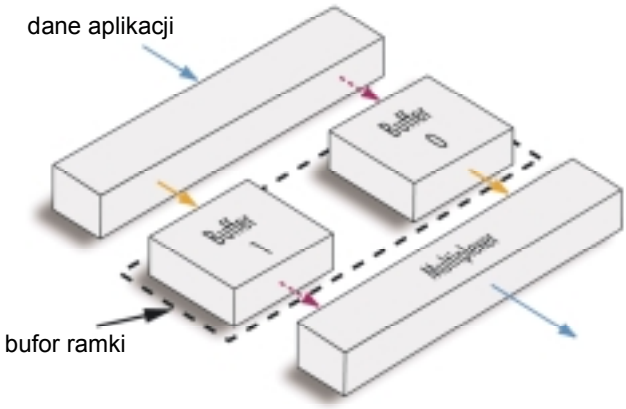
	Proste rysunki	Kreślenie i detalowanie	Modelowanie części 3D	Złożone powierzchnie	Styling	Złożenia części oraz analizy
Typ karty graficznej	2D	2D/3D	2D/3D[2]	3D	3D	3D
Rozdzielczość	≥800x600	≥800x600	≥1024x768	≥1024x768	≥1024x768	≥1024x768
Głębia koloru (color depth)	Low Color 256 kolorów (8 bit)	Low Color 256 kolorów (8 bit)	High Color 32K/64K kolorów (15-/16 bit)	True Color 16.7M kolorów (24 bit)	True Color 16.7M kolorów (24 bit)	True Color 16.7M kolorów (24 bit)
Przeźroczystość (stopień wykorzystania) (translucency)	N/A	N/A	niski	średni	średni	Wysoki
Z – buffering	N/A	N/A	16 bit	16- to 24 bit	24 bit	24- to 32 bit
Double buffering	Może[3]	Może[3]	TAK	TAK	TAK	TAK
3D texturing	N/A	N/A	TAK ≥4 MB	TAK ≥16 MB	TAK ≥16 MB	TAK ≥32 MB
Akceleracja geometrii (stopień zapotrzebowania) (geometry acceleration)	N/A	N/A	niski	średni	średni	wysoki

1. Kolumna Proste rysunki zwiera właściwości kart wymagane przy rysunkach 2D.
2. Większość programów typu CAD wykorzystuje rysunki 2D jako bazę dla modelowani 3D. Innymi słowy, 50% modelowania elementów 3D odbywa się w widokach typu 2D.
3. **Aplikacje typu 2D wymagają podwójnego buforowania (double buffering) tylko wówczas gdy przeprowadzane są pewne formy animacji 2D.**

Tab. 2 *Cechy kart graficznych w stosunku do wymagań aplikacji typu CAD/CAE/CAM*

DOUBLE BUFFERING

Double Buffering jest technologią stosowaną do płynnego i interaktywnego wyświetlania obrazu wysokiej jakości. Przykładem może być sytuacja w której użytkownik wydaje polecenie w aplikacji typu ***MCAD*** by poddany wcześniejszej wizualizacji obiekt płynnie obracać by dostrzec ewentualne wady, wtedy właśnie generowane są sekwencje klatek animacji. W takim przypadku szybkość żadnego obecnie znanego akceleratora geometrii 3D okazuje się niewystarczająca by wyświetlić za każdym razem nowo wygenerowaną klatkę, ponieważ wrażenie obserwatora byłoby takie, że stara klatka byłaby zamazywana przez nową klatkę. Aby rozwiązać taki problem „bufor ramki" (***frame buffer***) składa się z dwóch części, które odpowiadają buforowi nr 0 i buforowi nr 1. Zatem jak widać na Rysunku 4 „bufor ramki" składa się z bufora zer i bufora jedynek.



Rys. 4 *Bufor ramki składa się właściwie z dwóch buforów (zer i jedynek)*

Pierwsza wyświetlana klatka zapisywana jest w buforze (0) zerowym. Następnie podsystem graficzny wybiera sposób wyświetlenia klatki (poprzez sterownik programowy karty) na ekranie monitora, podczas kiedy równocześnie zapisywana jest kolejna klatka w buforze (1) jedynek. Karta graficzna wyświetla więc zawartość bufora (1) a do bufora (0) składowana jest następna klatka i tak dalej.
Innymi słowy wszystkie klatki oznaczone liczbami nieparzystymi (1, 3, 5, 7, ...) są zapisywane i wyświetlane z bufora (0), podczas gdy klatki parzyste (2, 4, 6, 8, ...) są zapisywane i wyświetlane z bufora (1). Dwa wymienione bufory są bezpośrednio powiązane z buforem frontowym i buforem wstecznym (multipleksery).
Dlatego ***Double Buffering*** oznacza de facto, że podsystem graficzny „przeskakuje" pomiędzy dwoma buforami wyświetlając obraz z wysoką częstotliwością, taką która sprawia, że oko ludzkie odbiera wyświetlane informacje w sposób płynny. Należy zwrócić uwagę, iż w przypadku akceleratorów 3D technika wyświetlania ***Double Buffering*** jest powiązana również z każdym wyświetlanym obrazem ***RGB*** na płaszczyźnie.
Uwaga: W związku z koniecznością zapewnienia większej pamięci przeznaczonej dla segmentu drugiego bufora niektóre karty graficzne umożliwiają wyświetlanie przy wykorzystaniu techniki ***Double Buffering*** osiąganych w niskich rozdzielczościach lub przy niskiej „głębi kolorów" co implikuje drastyczne zmniejszenie jakości wyświetlanego obrazu.

3D TEXTURING – Texturowanie

Podczas tworzenia obiektów w aplikacjach 3D można je traktować jako jednorodne bloki materiału powiedzmy plastiku. Dlatego właśnie możemy zadecydować by ich opis był uproszczony tzn. taki gdzie nadany materiał będzie jednolitym kolorem lub żeby ten sam obiekt był uformowany poprzez materiał teksturowany np. marmur. Możliwość nadawania obiektom tekstur pozwala w znacznym stopniu podnieść estetykę i stworzyć realistyczny obraz. Jedną z technik pozwalających na opisywanie obiektów teksturami jest tzw. „mapowanie tekstur" (***texture mapping***). W skrócie technika ta polega na stworzeniu obrazu 2D nazywanego teksturą i „pomalowanie" tym obrazem obiektu 3D.
Istnieje wiele algorytmów pozwalających na nakładanie dwuwymiarowych tekstur na obiekty 3D. Najmniej wyrafinowanymi są, próbkowanie punktowe lub próbkowanie najbliższego piksela. W średniej klasie

algorytmów można zaliczyć algorytm interpolacji podwójnej (***bilinear interpolation***), a de facto standardem dla wysokiej jakości obrazów jest algorytm interpolacji potrójnej (***trilinear interpolation***).
Nakładanie tekstury na elementy znajdujące się w głębi sceny jest operacją stosunkowo prostą. Jednak w przypadku dużych obiektów posiadających stosunkowo duże gabaryty powoduje specyficzne problemy polegające na tym iż nakładanie tekstury na znajdujące się w głębi elementy obiektów 3D przekłada, się na jej znaczny skok -ziarnistość. Rezultatem jest „uziarnienie" obiektów 3D wraz z oddalaniem się obiektu od obserwatora. Takie niepożądane efekty można zaobserwować nawet w przypadku statycznej sceny, a więc takiej w której obiekty są nieruchome. Wady te stają się jeszcze bardziej wyraźne w scenach dynamicznych ponieważ „oddalone" piksele są stale nakładane na wyświetlaną teksturę co przejawia się w migotaniu pomiędzy kolorami. Należy nadmienić że wada ta występuje w algorytmach punktowego próbkowania i podwójnej interpolacji nakładania tekstur.
Interpolacja potrójna jest algorytmem znacząco zwiększającym ilość wykonywanych obliczeń, w odniesieniu do dwóch wcześniej wymienionych, co przejawia się płynnym przedstawieniem dynamicznej sceny, bez migotania nawet dla obiektów bardzo oddalonych od punktu obserwacji.

Rys. 5 *Tekstura 2D „nakładana" jest niejako na obiekt 3D.*

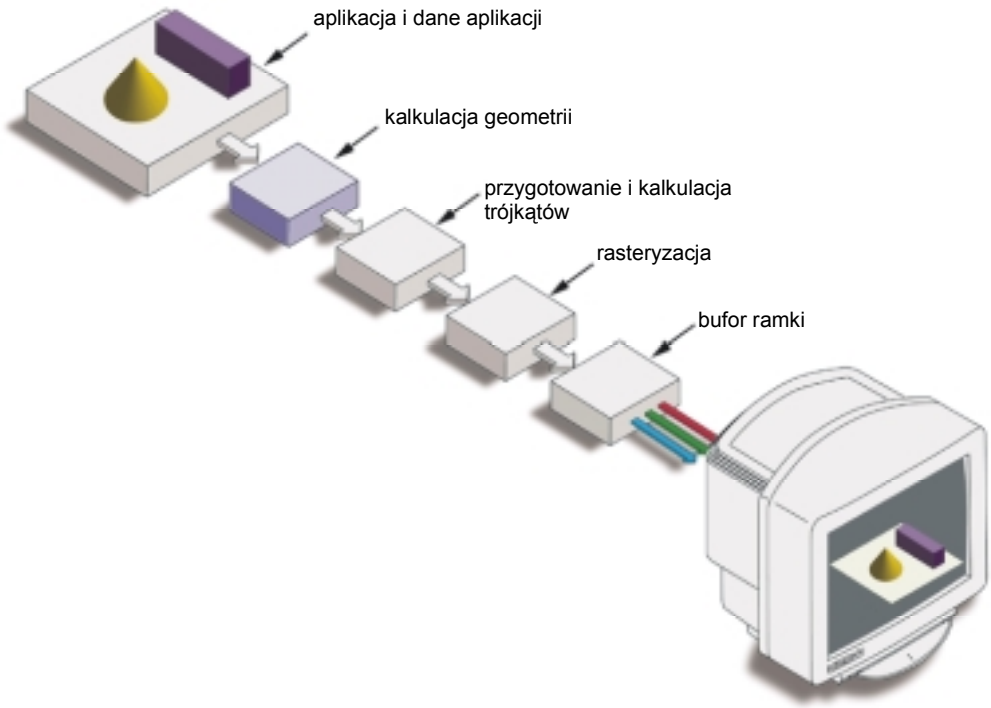
AKCELERACJA GEOMETRII

Rendering 3D jest graficznym przedstawieniem matematycznego opisu sceny 3D i przekonwertowanie jej na wyświetlany obraz 2D. W rezultacie obraz 2D wyświetlany na monitorze wygląda jak gdyby posiadał trzeci wymiar – głębię.
Obecnie znanych jest kilka technik renderingu do których zalicza się:

- cieniowanie płaskie (***flat shading***)
- cieniowanie metodą Gourard'a (***Gourard shading***)
- cieniowanie Phong (***Phong shading***)
- śledzenie (***ray traycing***)
- ***radiosity***

W dyskusji skoncentrujemy się na najczęściej wykorzystywanym algorytmie, cieniowaniu ***Gourard'a***. Pierwszym krokiem procesu jest kalkulacja geometrii. Kalkulacja geometrii zawiera w sobie proces transformacji świateł oraz transformację sceny. Każdy obiekt 3D jest właściwie reprezentowany przez zbiór wielokątów (zazwyczaj trójkątów). Proces transformacji rozpoczyna się od przekonwertowania współrzędnych $X_0Y_0Z_0$ „przestrzeni obiektu" dla każdego wierzchołka w każdym trójkącie na ich współrzędne $X_eY_eZ_e$, które są tzw. współrzędnymi „przestrzeni oka" (gdzie „przestrzeń oka" uwzględnia wszystkie obiekty w widocznym obszarze sceny, z punktu obserwacji). Proces transformacji zawiera także dalszą konwersję w której wartości współrzędnych $X_eY_eZ_e$ „przestrzeni oka" dla każdego wierzchołka są przeliczane na wartości $X_SY_SZ_S$ będącymi współrzędnymi ekranowymi.
Jeżeli umieszcza się obiekt w przestrzeni trójwymiarowej, jego cechy materiałowe takie jak kolor, odblask, półprzeźroczystość, muszą być odniesione do zadanego źródła światła. Proces oświetlenia jest realizowany w taki sposób, że obliczane są wartości RGB będące rezultatem padającego promienia świetlnego na pojedynczy wierzchołek w kolejności wyświetlane na ekranie monitora. Ostatecznie proces podziału na kolejne klatki (***clipping proces***) odrzuca te elementy (lub ich części) znajdujące się poza ekranem ze względu na to, że są one niewidoczne dla obserwatora.

Kalkulacja geometrii jest operacją wymagającą dużych mocy obliczeniowych, tak że wymagane jest wykonanie do 80 operacji zmiennoprzecinkowych dla jednego wierzchołka każdego trójkąta. Pomijając aplikacje 3D znajdujące swoją programową (***software***) implementację oraz bufor ramki implementowany zawsze sprzętowo pozostała operacje grafiki 3D mogą być realizowane zarówno drogą programowa jak i sprzętową. Jednakże dedykowane rozwiązania sprzętowe oferują daleko lepszą wydajność niż implementacje typu ***software***. Zastosowanie sprzętowej akceleracji geometrii pozwala użytkownikowi na płynne manipulowanie, obracanie obiektami i dużymi złożeniami na ekranie monitora bez strat jakości wyświetlanego obrazu 3D.



Rys. 6 *Kalkulacja geometrii jest pierwszym etapem w procesie renderingu.*

W przeszłości na każdym poziomie obróbki grafiki 3D znane były wyłącznie techniki programowe, służące do wizualizacji świata 3D. W ostatnich latach nastąpił gwałtowny rozwój specjalizowanych procesorów zapewniających sprzętowe wsparcie grafiki rastrowej z następującymi po nich akceleratorami trójkątów i stosowanych obecnie w kartach typu ***high-end*** akceleratorach geometrii. Zastosowanie akceleratorów geometrii w wyświetlaniu grafiki 3D pozwala na olbrzymie zwiększenie wydajności wyświetlania grafiki poprzez przejęcie wszystkich obliczeń numerycznych związanych z przetwarzaniem scen 3D i tym samym powoduje to odciążenie CPU komputera. Jednak akceleratory geometrii są bardzo skomplikowane gdzie wymagana jest zaawansowana technologia wytwarzania, co pociąga za sobą ich cenę

PODSUMOWANIE

Wybór nowego komputera oraz podsystemu graficznego nigdy nie jest sprawą łatwą, a urasta on do wagi problemu w przypadku uwzględnienia wszelkich różnic. Ten krótki opis funkcji akceleratorów 3D miał na celu przedstawienie niektórych najważniejszych wspólnych cech kart graficznych, aczkolwiek istnieje więcej aspektów odnoszących się do technologii 3D, których nie jesteśmy przytoczyć na łamach tego artykułu.
Wszystkich, którzy chcieliby dowiedzieć się więcej na temat technologii 3D Intergraph Computer Systems oferuje książkę „***Graphics Supercomputing on Windows NT***" opisującą aspekty grafiki 2D oraz 3D w zrozumiały i interesujący sposób.
Dla aplikacji typu ***MCAD*** moc oraz wydajność podsystemu graficznego jest zawsze sprawą krytyczną. Z tego też powodu użytkownik powinien zawsze testować aplikacje pozwalające na pracę na obiektach 3D w czasie rzeczywistym i do nich dobierać podsystem graficzny tak by spełniał wszystkie stawiane przed nim wymagania.

Z Cilve'm można się kontaktować na stronie www.maxmon.com

Tłumaczenie: Wszebor Boksa

Zredagowano i przetłumaczono za zgodą ***Helmers Publishing, Inc.*** *Artykuł ukazał się w „****Desktop Engineering Magazine****", Aug. 1998,* www.deskeng.com

Ilustracje pochodzą z książki „***Graphic Supercomputing on Windows NT***", Intergraph Computer Systems, 1998