

P. 1847/91



3 1991

informatyka

Symulatory lotu

Miesięcznik
ISSN 0542-9951
INDEKS 36124

KOLEGIUM REDAKCYJNE:

mgr Jarosław DEMINET
dr inż. Wacław ISZKOWSKI
mgr Teresa JABŁOŃSKA
(sekretnarz redakcji)
Władysław KLEPACZ
(redaktor naczelny)
dr inż. Wojciech MOKRZYCKI
mgr inż. Jan RYZKO
mgr Hanna WŁODARSKA

PRZEWODNICZĄCY RADY PROGRAMOWEJ:

Prof. dr hab.
Juliusz Lech KULIKOWSKI

WYDAWCA:

Wydawnictwo Czasopism i Książek
Technicznych SIGMA NOT
Spółka z o.o.
ul. Biała 4
00-950 WARSZAWA
skrytka pocztowa 1004

Redakcja:

01-552 Warszawa,
Pl. Inwalidów 10, p. 104, 105
tel. 39-14-34

Materiałów nie zamówionych
redakcja nie zwraca

SKŁAD I DRUK:

SIGMA-NOT Sp. z o.o. z 583/90
Obj. 3,0 ark. druk.
Cena egzemplarza 12 000 zł

**W sprawach ogłoszeń
prosimy zwracać się
bezpośrednio
do Redakcji**

W numerze:

	Strona
Wizualizacja scen dla potrzeb symulatorów lotu – Jan Zabrodzki	1
Struktura wieloprocessorowa dla celów cyfrowego generowania obrazu – Jerzy Chrzyszcz	5
SDeL – język opisu scen geometrycznych – Andrzej Paják	8
Układ szybkiego wypełniania pamięci obrazu – Jerzy Chrzyszcz	11
Algorytm rysowania barwnego odcinka i jego wykorzystanie do cieniowania wnętrza trójkąta – Janusz Rzeszut	14
Sposób odwzorowania światła na ekranie monitora graficznego w animowanych scenach nocnych – Cezary Stepień, Tomasz Prokop	17
Efekt perspektywy powietrznej w generowaniu obrazu dla symulatorów – Janusz Rzeszut	19
Realizacja efektu zmiany koloru obiektu w ośrodku optycznym w systemie generowania obrazu w czasie rzeczywistym – Grzegorz B. Mazur	22

W najbliższych numerach:

- Tomasz Berus i Bohdan Macukow charakteryzują rozwój badań w dziedzinie modelowania i budowy sieci neuronowych, a także omawiają koncepcję neurokomputera i porównują go z komputerem klasycznym.
- Dorota Kuchta w pierwszej części artykułu o systematycznym testowaniu programów charakteryzuje i podaje przykłady najczęściej stosowanych metod takiego testowania.
- Leszek Nowaczyk omawia podstawowe pojęcia programowania obiektowego.

Warunki prenumeraty

Przyjęcie prenumeraty – wyłącznie na podstawie dokonanej wpłaty na drukach dostarczanych dotychczasowym prenumeratorom przez Wydawnictwo, lub nowym – po uprzednim zgłoszeniu zapotrzebowania (pisemnie lub telefonicznie) w Zakładzie Kolportażu Wydawnictwa.

Blankiet wpłaty – powinien zawierać następujące informacje: dokładna nazwa i adres (z kodem pocztowym) zamawiającego, tytuły zamawianych czasopism, ich liczbę i okres prenumeraty.

Wpłata – zgodnie z podanymi cenami należy dokonać w banku lub w UPT na konto podane na naszym blankiecie, tj:

Państwowy Bank Kredytowy III O/Warszawa nr: 370015-1573-139-11

Prenumeratory zbiorowi – osoby prawne obowiązują blankiety „Wpłata-Zamówienie”. Cena normalna.

Prenumeratory indywidualni – osoby fizyczne obowiązują blankiety typu przekazy dla wpłat na rachunki bankowe. Cena normalna.

Prenumerata ulgowa – zgodnie z podaną ceną ulgową przysługuje wyłącznie osobom fizycznym, będącym członkami SNT, studentom i uczniom szkół zawodowych. Uczniowie szkół ogólnokształcących mogą zamówić w prenumeracie ulgowej tylko miesięcznik „Aura”

Uwaga! w podanym okresie prenumeraty można zamówić tylko po jednym egzemplarzu z każdego tytułu.

Prenumerata ze zleceniem wysyłki za granicę – cena prenumeraty ze zleceniem wysyłki za granicę jest dwukrotnie wyższa od ceny normalnej.

Należy podać dokładny adres odbiorcy za granicą.

Terminy przyjmowania prenumeraty:

- do 10 listopada na I, II, III, IV kwartał następnego roku
- do 28 lutego na II, III, IV kwartał br.
- do 31 maja na III i IV kwartał br.
- do 31 sierpnia na IV kwartał br.

Zmiany w prenumeracie, np. zmiana liczby tytułów, liczby egzemplarzy, rezygnacja z prenumeraty, można zgłaszać tylko w podanych terminach z mocą obowiązującą od następnego kwartału.

Egzemplarze archiwalne (z lat ubiegłych)

Można nabyć za gotówkę w Klubie Prasy Technicznej, Warszawa, ul. Mazowiecka 12 (tel. 26-80-16) lub zamówić pisemnie w Zakładzie Kolportażu, Dział Handlowy, 00-950 Warszawa, skr. poczt. 1004 (tel. 40-37-31), na rachunek lub za zaliczeniem pocztowym.

Informacji o prenumeracie udziela: Zakład Kolportażu Wydawnictwa SIGMA-NOT Spółka z o.o., 00-716 Warszawa, ul. Bartka 20, skr. 1004. Telefony: 40-00-21 wewn. 293, 295, 299 lub 40-30-86, 40-35-89.

Wstępna cena jednego egzemplarza na 1991 rok: normalna – 12 000 zł, ulgowa – 8 400 zł

Wartość prenumeraty (w zł):

Normalna: kwartalna – 36 000, półroczna 72 000, roczna 144 000

Ulgowa: kwartalna – 25 200, półroczna 50 400, roczna 100 800

Uwaga: W przypadku zmiany cen w okresie objętym prenumeratą, prenumeratorzy zobowiązani są do dopłaty różnicy cen.



Wizualizacja scen dla potrzeb symulatorów lotu

Jednym z ważnych zastosowań grafiki komputerowej jest cyfrowe generowanie obrazów dla potrzeb symulatorów lotu. W artykule przedstawiono podstawowe problemy występujące przy budowie systemów wizualizacji takich symulatorów oraz stosowane rozwiązania. Przedstawiono również główne założenia systemu wizualizacji SLOT-A realizowanego w Instytucie Informatyki PW.

Od dawna są znane zalety stosowania symulatorów lotu do szkolenia pilotów. Efekty szkoleniowe w dużym stopniu zależą od budowy symulatora i od wierności odtwarzania różnych efektów, z którymi spotyka się pilot w czasie rzeczywistego lotu. Wśród wielu różnych efektów podlegających symulacji, jednymi z ważniejszych są efekty oddziałujące na wzrok pilota. Dlatego prawie wszystkie nowoczesne symulatory są wyposażone w systemy wizualizacji, których zadaniem jest wytworzenie efektów wzrokowych możliwie wiernie oddających to, co widzi pilot z kabiny samolotu.

Mimo dużego znaczenia wrażeń wzrokowych historia rozwoju systemów wizualizacji nie jest zbyt długa. Pierwsze próby symulacji wrażeń wzrokowych podjęto około 25–30 lat temu; polegały one na zastosowaniu techniki filmowej. Oczywiście wadą takiego rozwiązania były duże koszty uzyskania filmu odpowiedniego do celów szkoleniowych, a także możliwość trenowania jedynie przy jednym ustalonym torze przelotu. Metody tej zaniechano z chwilą, gdy pojawiła się możliwość zastosowania techniki telewizyjnej pracującej w systemie zamkniętym (połowa lat sześćdziesiątych). Idea tej metody polegała na tym, że symulowanym zmianom położenia samolotu w czasie ćwiczenia towarzyszyły zmiany położenia kamery telewizyjnej, która poruszała się nad przestrzennym modelem terenu. Obraz z kamery był wyświetlany na monitorze umieszczonym przed kabiną pilota. Zaletą takiego rozwiązania jest niewątpliwie duża wierność odtworzenia terenu, nad którym odbywa się lot, oraz możliwość poruszania się w dowolnym kierunku nad powierzchnią makiety. Do wad należy zaliczyć duże koszty realizacji systemu oraz ograniczenie się do jednego terenu ćwiczeń, a także brak możliwości symulacji różnych zjawisk atmosferycznych.

Metodę tę stosowano do połowy lat siedemdziesiątych (niektóre z takich systemów są wykorzystywane do dzisiaj), kiedy to pojawiła się możliwość wykorzystania metod cyfrowego generowania obrazów i wyświetlania ich na monitorze komputera. Pierwsze komputerowe systemy wizualizacji umożliwiły generowanie prostych scen nocnych bądź terenu o postaci szachownicy. W miarę upływu czasu opracowywane systemy o coraz większych możliwościach symulowania wrażeń wzrokowych.

Cechy funkcjonalne systemów wizualizacji

Zasadniczym celem stosowania symulatorów lotu jest uzyskanie odpowiedniej efektywności szkolenia. Z tego punktu widzenia podstawową cechą systemu wizualizacji jest to, by dostarczał on wrażeń wzrokowych umożliwiających pilotowi orientację w przestrzeni przy spełnieniu warunku, że nie wystąpią zakłócenia związane z wrażeniami sprzecznymi (bądź rozbieżnymi) z obserwacjami świata rzeczywistego. Stwierdzenie to jest o tyle istotne, że naturalne jest dążenie do uzyskania jak największego realizmu generowanych scen. Teoretyczne wrażenia wzrokowe pilota znajdującego się w symulatorze nie powinny się różnić od wrażeń wzrokowych pilota w rzeczywistym samolocie. W praktyce względy technologiczne nie pozwalają uzyskać takich efektów. Dlatego zawsze występuje konieczność zrezygnowania z realizacji pewnych efektów i przyjmowania rozwiązań kompromisowych.

Podana ogólna zasada niejednokrotnie pozwala dokonać właściwego wyboru. Najczęściej chodzi o wybór między realizmem sceny a czasem jej generowania. W grafice komputerowej są przecież znane metody umożliwiające uzyskanie obrazu o wręcz fotograficznej dokładności; jednak uzyskanie takiego obrazu trwa kilka lub kilkadziesiąt godzin.

Po tych uwagach ogólnych można sformułować szczegółowe wymagania, jakie powinien spełniać system wizualizacji, żeby mógł dostarczać wrażeń wzrokowych zbliżonych do tych, których doznaje pilot w kabinie samolotu. Ograniczymy się tu do wrażeń dostarczanych przez otoczenie samolotu widziane przez okna w kabinie samolotu.

System wizualizacji powinien zapewniać możliwość obserwacji otoczenia we wszystkich fazach lotu, a więc w czasie postoju na lotnisku, kołowania na pas startowy, startu, trwania lotu oraz w czasie lądowania. Realizacja tych wymagań wiąże się z koniecznością reprezentowania obiektów płaskich i przestrzennych obserwowanych zarówno w stanie statycznym, jak i przy zmiennej prędkości poruszania się względem tych obiektów. Obiekty te muszą być ponadto odtwarzane w warunkach obserwacji z różnej odległości i z różnych kierunków.

Wśród obiektów, które podlegają symulacji, występują obiekty różnych klas. Są to obiekty naturalne, takie jak pola, lasy, rzeki, góry, drzewa oraz obiekty będące dziełem człowieka, takie jak pas startowy, budynki, wieże, światła. Oprócz obiektów stałych, których położenie jest ściśle określone, może występować konieczność reprezentacji obiektów ruchomych, na przykład poruszającego się po lotnisku autobusu lub innego kołującego samolotu.

Pilot powinien mieć możliwość obserwowania obiektów w różnych warunkach pogodowych (mgła, zachmurzenie, deszcz, śnieg, słoneczna pogoda itp.) oraz o różnych porach dnia (południe, zmierzch, noc) i roku (zima, lato).

System wizualizacji powinien zapewnić możliwość wykonywania ćwiczeń nad różnymi terenami o dużych powierzchniach (tereny zurbanizowane, pustynne, górskie itp.). Powinien on również umożliwiać prowadzenie ćwiczeń związanych ze startem i lądowaniem na różnych lotniskach.

W dobrym symulatorze lotu pilot powinien mieć możliwość wykonywania dowolnych manewrów dopuszczalnych przez symulowany samolot. Dlatego system wizualizacji powinien działać z taką szybkością i zapewnić takie pole obserwacji, jak to wynika z możliwości technicznych samolotu. W szczególności na przykład powinno być możliwe reprezentowanie scen przy locie poziomym, ukośnym lub nawet pionowym w kierunku ziemi bądź nieba. Przy największych dopuszczalnych zmianach prędkości, kierunku bądź któregoś z kątów charakteryzujących tor lotu, pilot powinien mieć zapewnione wrażenie ciągłości obserwowanych scen.

Problemy realizacyjne

Wymienione wymagania stawiane systemom wizualizacji symulatorów lotu pozwalają uzmysłowić sobie, jak wiele problemów należy rozwiązać przy realizacji systemu wizualizacji. Są to zarówno problemy natury algorytmicznej, jak i problemy związane z realizacją sprzętową i z opracowaniem oprogramowania dla systemu.

Wśród wymienionych wymagań można wyróżnić dwie klasy – te, które dotyczą złożoności pojedynczej sceny oraz te, które są związane z dynamiką prezentacji kolejnych scen. Złożoność pojedynczej sceny decyduje o wierności odzwierciedlenia rzeczywistości, dynamika prezentacji scen decyduje natomiast o uzyskaniu efektu ciągłości zmian i odzwierciedlenia wrażeń związanych z dynamiką lotu (prędkość, przyspieszenie, zmiana kierunku lotu, inne obiekty ruchome, zmieniające się warunki pogodowe, zmiana warunków oświetlenia itd.).

Zacznijmy od zagadnień związanych z realizacją pojedynczej sceny.

W systemach wizualizacji wyróżnia się z reguły dwie części: bazę danych o reprezentowanym terenie oraz system generowania obrazu. Baza danych zawiera informację o całym terenie, nad którym może odbywać się lot. Są tam zawarte informacje o wszystkich obiektach występujących na danym terenie, jak też o ich położeniu we współrzędnych globalnych. Informacja o obiekcie zawarta w bazie powinna być na tyle obszerna, by na jej podstawie było możliwe wygenerowanie obrazu obiektu obserwowanego z wybranego punktu przestrzeni. Baza danych o terenie jest przygotowywana oddzielnie (off-line).

Dla określonego położenia samolotu w przestrzeni (na lotnisku bądź nad terenem ćwiczenia) wyznaczonego przez sześć współrzędnych (trzy współrzędne geograficzne x , y , z oraz trzy współrzędne kątowe – pochylenie, przechylenie, azymut) jest generowana scena. Proces generowania zaczyna się od wybrania z bazy danych tych obiektów, które znajdują się w polu widzenia pilota. Z kolei informacje o wybranych obiektach są poddawane odpowiednim przekształceniom tak, by w efekcie uzyskać na ekranie rzut perspektywiczny obserwowanego fragmentu terenu.

Ze względu na łatwość i szybkość wykonywania odpowiednich obliczeń w rzeczywistych systemach wizualizacji stosuje się wielokątową reprezentację obiektów 2- i 3-wymiarowych (2W i 3W); światła są reprezentowane jako punkty odpowiednio rozmieszczone w terenie. Dzięki temu jest możliwe wykonywanie wielu wstępnych obliczeń jedynie w odniesieniu do wierzchołków wielokątów i punktów reprezentujących światła (transformacja układu współrzędnych, obcinanie, rzutowanie). Dopiero w końcowej fazie obliczeń następuje wypełnianie powierzchni wielokątów, rozwiązywanie problemu przesłaniania obiektów, generowanie obrazów światła. Zależnie od złożoności systemu wizualizacji powierzchnie wielokątów są wypełniane jednym kolorem, cieniowane z uwzględnieniem odległości, oświetlenia, pory dnia itp., bądź też są wypełniane odpowiednią teksturą.

Złożoność pojedynczej sceny jest na ogół charakteryzowana przez określenie dopuszczalnej liczby wielokątów oraz liczby światła w scenie. Przyjmuje się, że średniej klasy systemy powinny zapewnić możliwość generowania sceny składającej się z ponad 100 wielokątów i około 1000 światła. Liczba bezpośrednio dostępnych kolorów nie powinna być mniejsza niż 2^{15} .

Istotnym problemem przy generowaniu poszczególnych scen jest uwzględnienie zmiany odległości, z której ta scena jest obserwowana. Na przykład w sytuacji, gdy samolot znajduje się na pasie startowym istotne jest wierne odwzorowanie pasa startowego wraz z jego oznakowaniem, fakturą powierzchni i oświetleniem, wieży kontrolnej, budynków itp. oraz uwzględnienie efektów przesłaniania. W przypadku sceny widzianej z dużej wysokości powierzchnia widzianego terenu jest znacznie większa. Większa jest również liczba obiektów, które potencjalnie znajdują się w polu widzenia. Zmienia się jednak obraz poszczególnych obiektów. Niektóre są tak małe, że przestają być w ogóle widoczne, inne są widziane jedynie w ogólnym zarysie, bez szczegółów z nimi związanych (na przykład pas startowy). Mniejsze jest znaczenie odwzorowania obiektów 3W wytworzonych przez człowieka, istotne jest natomiast odwzorowanie obiektów naturalnych, takich jak lasy, pola, góry itp. Jeszcze inny rodzaj elementów składających się na scenę wystąpi w przypadku lotu nad chmurami bądź przy locie wznoszącym się. Należy jednak pamiętać, że każdorazowo złożoność sceny nie może przekroczyć dopuszczalnej liczby wielokątów lub zestawów światła.

Tak więc przy generowaniu pojedynczej sceny istotne jest rozwiązanie takich problemów, jak wybór jednolitej reprezentacji obiektów, przy bardzo dużej różnorodności tych obiektów, oraz zapewnienie w miarę stałej złożoności scen, przy obserwacji scen z różnych punktów przestrzeni i w różnych warunkach meteorologicznych. Dodatkowe ograniczenie w procesie generowania sceny jest związane z czasem, który może być na ten cel poświęcony. Czas ten wynika z wymagań związanych z dynamiką prezentacji scen.

Pilot w każdej chwili lotu musi mieć wrażenie, że obraz, który ogląda, jest obrazem ciągłym. Oznacza to, że kolejne sceny muszą być wyświetlane na ekranie monitora z częstotliwością 50–60 Hz. Chodzi tu oczywiście o uniknięcie efektu migotania obrazu. Czasami dopuszcza się, by każda scena była wyświetlana dwukrotnie, wtedy kolejne sceny są generowane z częstotliwością 25–30 Hz.

Dopuszczając w symulatorze pełną swobodę manewrów samolotem trzeba liczyć się z koniecznością przyjęcia założenia, że dwie kolejne sceny nie są ze sobą w żaden sposób powiązane i muszą być generowane całkowicie niezależnie od siebie i zawsze od początku. Widać tu istotną różnicę w stosunku do animacji, gdzie istnieją powiązania między kolejnymi scenami.

Przy wyświetlaniu sekwencji scen nie mogą pojawiać się efekty uboczne, które mogą być wynikiem generowania scen w technice

rastrowej. Efekty takie mogłyby po pierwsze rozpraszać uwagę trenującego w symulatorze, a po drugie dostarczałyby wrażeń wzrokowych, z którymi pilot nie spotyka się w rzeczywistości.

W czasie wyświetlania określonej sceny nie powinna ona ulegać zmianom. Można to rozwiązać, stosując podwójną pamięć obrazu – wtedy gdy do jednej pamięci są zapisywane informacje o nowo tworzonej scenie jest wyświetlana scena zapisana wcześniej w drugiej pamięci. Dalej, konieczne jest eliminowanie efektów typu „zabkowane linie” oraz ginące i pojawiające się niewielkie obiekty. Zwalczanie tego rodzaju zakłóceń wiąże się z koniecznością zwiększania liczby i dokładności obliczeń. Z punktu widzenia dynamiki wyświetlanych scen ważne jest prawidłowe nakładanie informacji o obiektach ruchomych, które mogą znaleźć się w polu widzenia pilota w różnych fazach lotu. Podobnie prawidłowo muszą być uwzględnione sekwencje zmian światła występujących na lotniskach.

Analiza możliwości technicznej realizacji systemów wizualizacji pokazuje, że wymagają one stosowania specjalizowanych systemów komputerowych. Dostępne systemy uniwersalne nie pozwalają uzyskać systemów wizualizacji o zadowalającej jakości.

W każdym systemie wizualizacji można wyróżnić cztery podstawowe bloki połączone ze sobą szeregowo: blok tworzenia sceny, blok przekształceń geometrycznych, blok tworzenia obrazu oraz monitor.

Zadaniem pierwszego bloku jest przygotowanie informacji wyjściowej do utworzenia kolejnej sceny. Chodzi tu o wybór odpowiednich obiektów z bazy danych i wstępne przetwarzanie informacji tak, by była ona bezpośrednio dostępna dla następnego bloku. W publikacji [10] szacuje się, że zależnie od złożoności systemu wizualizacji moc obliczeniowa potrzebna do realizacji zadań pierwszego bloku wynosi od 1 do 40 MIPS. Rodzaj wykonywanych przez ten blok obliczeń pozwala stosować do jego realizacji odpowiednio szybkie komputery uniwersalne.

Blok drugi ma zadanie wykonania wszelkich przekształceń geometrycznych związanych z transformacją perspektywiczną, obcinaniem, rzutowaniem itp. oraz innych obliczeń (np. gradienty parametrów geometrycznych i kolorów) niezbędnych do wytworzenia pełnej informacji potrzebnej dla bloku następnego. Wydajność tej części systemu jest określana często przez podanie liczby wielokątów, które mogą być przetworzone w jednostce czasu (lub liczby krawędzi; szacunkowo przyjmuje się, że przetworzenie trzech krawędzi jest równoważne przetworzeniu jednego wielokąta). Liczba przetwarzanych wielokątów waha się od kilkuset do kilku tysięcy w ciągu sekundy. Według autora pracy [10] realizacja zadań tego bloku za pomocą uniwersalnego komputera wymagałaby mocy obliczeniowej na poziomie 60 MIPS. Najczęściej blok ten jest realizowany za pomocą specjalizowanego sprzętu o architekturze równoległej.

Zasadniczym zadaniem bloku trzeciego jest określenie koloru każdego piksla przy uwzględnieniu efektów oświetlenia, warunków atmosferycznych, odległości, tekstury, efektów odskłaniania obrazu itd. W pracy [10] wymagana na tym etapie moc obliczeniowa uniwersalnego komputera szacuje się na 10 000 MIPS. Blok ten jest realizowany zawsze za pomocą specjalizowanych rozwiązań układowych.

Produkcja systemów wizualizacji zajmuje się jedynie niewielką liczbą firm na świecie. Natomiast systemy wizualizacji, w których wykorzystuje się metody cyfrowego generowania obrazów, były produkowane przez największe firmy wytwarzające symulatory lotów (General Electric, Singer/Link, Evans and Sutherland, McDonnell Douglas). Ceny systemów wizualizacji dostarczanych przez te firmy kształtowały się na poziomie kilku milionów dolarów (wersje wojskowe do 10 mln dolarów). Dla ilustracji, parametry jednego z większych systemów Image III (1983 r.) firmy Singer-Miles (wykorzystywanych na przykład do szkolenia pilotów samolotu Boeing 747) są następujące.

System Image III pozwala wyświetlać sceny dzienne z częstotliwością 50 Hz, a sceny nocne i o zmierzchu z częstotliwością 30 Hz. Rozdzielczość obrazu wynosi 1000×756 pikseli. Scena dzienna może zawierać do 256 wielokątów, przy ograniczeniu liczby boków do 15. Są dostępne 64 kolory, a ponadto wzory tekstur dla nieba i powierzchni ziemi oraz dodatkowo 8 innych wzorów tekstur. Scena może zawierać do 4096 różnego rodzaju punktów świetlnych (scena dzienna do 1000 punktów), przy czym do wyświetlania światła jest dostępnych 16 kolorów i 64 poziomy jasności. System może wyświetlać obrazy równocześnie w trzech kanałach (widok przez okno przednie i przez dwa boczne okna kabiny). Jest możliwe symulowanie zmiennych warunków pogodowych (widzialność, chmury, mgła, oświetlenie, stan powierzchni pasa startowego, śnieg). Istnieje możliwość symulowania obecności obiektów ruchomych (cysterna, autobus itp.).

Późniejszy system Image IV (z 1986 r.) umożliwia generowanie scen 3W o dużej złożoności z możliwością realizacji, tekstury i z uwzględnieniem

takich efektów, jak oświetlenie, przezroczystość, cienie, ruch, pogoda itp. Koszt tego systemu wynosi 1,9 mln dolarów; koszt systemu w wersji ograniczonej do scen nocnych wynosi natomiast 1,2 mln dolarów.

W ciągu ostatnich kilku lat pojawiły się dwie nowe firmy, Ivex oraz Hi-Tech Marketing, które oferują tańsze systemy. Koszt ich wyrobów jest na poziomie 0,5 mln dolarów. System Paragon II (Hi-Tech) umożliwia generowanie scen o złożoności do 10 000 wielokątów i 4000 świateł z uwzględnieniem mechanizmów odklócania. System VDS 1000 firmy Ivex umożliwia generowanie scen dziennych 2W z wypełnianiem powierzchni teksturą (rozdzielczość 640×480 przy 2^{24} kolorach/piksel). Światła (do 2000) są w tym systemie generowane opcjonalnie.

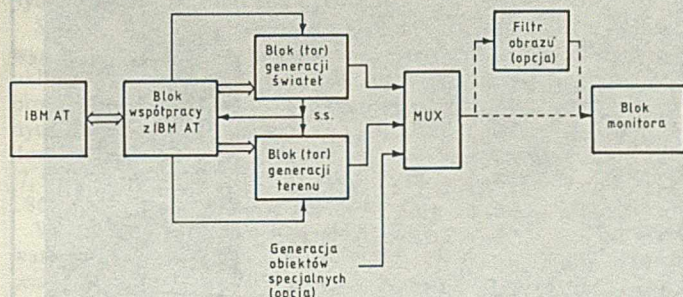
System wizualizacji SLOT-A

System SLOT-A jest realizowany z wykorzystaniem metody cyfrowego generowania obrazów. Założona rozdzielczość obrazu wynosi 640×480 pikseli. Liczba kolorów dostępnych bezpośrednio do wyświetlenia piksla wynosi 2^{15} . Częstotliwość generowania scen – 30 Hz, a częstotliwość wyświetlania obrazu 60 Hz.

Sceny są budowane z wielokątów o liczbie boków nie większej niż 8. Sceny mogą zawierać obiekty płaskie 2W (przyjmuje się, że ziemia jest płaska) oraz niewielką liczbę obiektów 3W (prostokątności, ostrosłupy). Na tle obrazu terenu mogą być wyświetlone światła związane z oświetleniem pasa startowego, lotniska, ulic. Zakłada się możliwość wyświetlania wszystkich rodzajów świateł związanych z oświetleniem lotniska. W szczególności można wyświetlać ciągi jednolitych świateł, światła kierunkowe, światła migające. Przyjęto, że światła znajdują się na poziomie ziemi.

System umożliwia generowanie scen dziennych, nocnych i o zmierzchu. Sposób generowania sceny jest jednolity dla każdej pory dnia. Zakładana złożoność sceny wynosi minimum 100 wielokątów i 1000 punktów świetlnych. Przy wyznaczaniu kolorów pikseli uwzględnia się wpływ odległości, przezroczystości powietrza, pory dnia itp.

W projekcie nie przyjęto żadnych ograniczeń dotyczących dopuszczalnych wartości kątów charakteryzujących położenie samolotu. Powierzchnia obszaru, nad którym może poruszać się samolot, wynosi $64 \text{ km} \times 64 \text{ km}$.



Na rysunku przedstawiono schemat blokowy systemu SLOT-A. W systemie tym mikrokomputer IBM PC/AT jest odpowiedzialny za obsługę operatora, współpracę z symulatorem, wybór obiektów do wyświetlania w kolejnej scenie, przygotowanie parametrów dla dalszych elementów systemu oraz danych do wyświetlania w postaci zbioru wielokątów określonych przez swoje wierzchołki i kolor oraz zbiór świateł. W dalszej, specjalizowanej części systemu po wykonaniu odpowiednich transformacji w odniesieniu do wierzchołków (blok generowania terenu) i do punktów świetlnych (tor generowania świateł) są określone ich końcowe pozycje na ekranie i jest wypełniana pamięć obrazu z zachowaniem zasad ciągłości kolorowania, z uwzględnieniem zadanych warunków pogodowych, pory roku, dnia itp. Przyjęto, że scena jest tworzona w pamięci obrazu na zasadzie nakładania na siebie kolejnych wielokątów i świateł. Oznacza to, że kolejne obiekty są wprowadzane do pamięci obrazu w kolejności odwrotnej do priorytetu ich widoczności. Skompletowana scena jest wyświetlana na ekranie monitora.

Przy przyjętych założeniach, że scena może być tworzona z ponad 100 wielokątów i 1000 punktów świetlnych, tor generowania obrazu musi zapewnić wykonanie przekształceń geometrycznych dla około 2000 punktów na jedną scenę, czyli dla około 60 000 punktów w ciągu sekundy. Natomiast zakładana szybkość generowania punktów obrazu wynosi około 9,2 M punktów/sekundę (szybkość wyświetlania jest dwa razy większa).

Tor generowania świateł i tor generowania terenu działają niezależnie. Wynikiem działania toru świateł jest scena zawierająca wyłącznie

światła, pamiętana w pamięci pomocniczej. Wynikiem działania toru generowania terenu jest scena zawierająca wyłącznie obraz terenu (i nieba), również pamiętana w pomocniczej pamięci. Na podstawie tych dwóch pomocniczych scen (i ewentualnie dodatkowej opcjonalnej sceny z wyróżnionymi obiektami specjalnymi) jest tworzona scena końcowa w pamięci obrazu (podwójnej). Do synchronizacji całego bloku generowania obrazu są wykorzystywane sygnały s.s. generowane w bloku generowania świateł.

* * *

Przy projektowaniu systemu SLOT-A przyjęto wiele założeń uproszczających, głównie ze względu na ograniczone możliwości zakupu sprzętu i elementów. Niektóre szczegółowe problemy napotkane przy projekcie systemu SLOT-A przedstawiono w pracach [1, 2, 5, 7, 8, 11, 12, 14]. Przewidywane modyfikacje systemu będą szły w kierunku zwiększenia rozdzielczości obrazu (do 1024×768 pikseli), zwiększenia liczby wielokątów tworzących scenę oraz dodania możliwości generowania tekstury wypełniającej powierzchnie wielokątów.

LITERATURA

- [1] Chrzyszcz J.: Struktura wieloprocessorowa dla celów generacji obrazu*
- [2] Chrzyszcz J.: Układy szybkiego wypełniania pamięci obrazu*
- [3] Deyo R., Briggs J.A., Doenges P.: Getting Graphics in Gear: Graphics and dynamics in driving simulation. Computer Graphics. Vol. 22, No. 4, pp. 317–326, 1988
- [4] Image III. Product description, Singer-Link/Miles.
- [5] Mazur G.B.: Realizacja efektu zmiany koloru obiektu wynikającej z własności optycznych ośrodka w systemie generacji obrazu w czasie rzeczywistym*
- [6] Moxon J.: Visuals for all. Flight International, pp. 39–41, 21 February 1987
- [7] Pajak A.: Prosty język dla definiowania scen geometrycznych*
- [8] Raczkowski J.: Mechanizm zarządzania pamięcią dla systemów grafiki komputerowej wymagających dostępu do dużych baz danych*
- [9] Ranjbaron S.E., Swallow R.J.: COMPUTROL in flight simulation. Eurographics pp. 321–329, 1980
- [10] Ralfe J.M., Staples K.J.: Flight Simulation. Cambridge University Press, 1988
- [11] Rudowski M.: Koncepcja reprezentacji danych o terenie oraz modyfikacji koloru obiektów i tła dla syntezy obrazów w czasie rzeczywistym*
- [12] Rzeszut J.: Efekt perspektywy powietrznej w generacji obrazu dla symulatorów*
- [13] Schachter B.J.: Computer Image Generation. John Wiley and Sons, N.Y. 1983
- [14] Stepien C., Prokop T.: Sposób odwzorowania świateł na ekranie monitora graficznego w animowanych scenach*
- [15] Wanstall B.: Combat crews to „fly” war routes? Interavia No. 3, pp. 271–273, 1976
- [16] Yan J.K.: Advances in Computer-Generated Imagery for Flight Simulation. IEEE Computer Graphics and Application, pp. 37–51, August 1985
- [17] Zyda M.J. et al.: Flight Simulation for under \$ 100000. IEEE Computer Graphics and Applications, pp. 19–28, January 1988.

* Grafika Komputerowa i Przetwarzanie Obrazów '90. Materiały I Krajowej Konferencji Instytutu Podstaw Informatyki PAN w Kazimierzu Dolnym, 21–24 maja 1990, Warszawa, 1990

Instytut Informatyki Politechniki Łódzkiej, Polskie Towarzystwo Cybernetyczne Oddział w Łodzi i Polskie Towarzystwo Informatyki Medycznej Oddział w Łodzi organizują w Łodzi w dniach 19–20 września 1991 r. II Krajową Konferencję

KOMPUTERY W MEDYCYNIE '91

Konferencja jest poświęcona prezentacji prac naukowo-badawczych dotyczących wykorzystania komputerów w medycynie. Jej celem jest wymiana doświadczeń zarówno w zakresie poszukiwania nowych koncepcji systemów informatycznych, jak i eksploatacji systemów już wdrożonych.

Program konferencji obejmuje:

- ☐ systemy informatyczne w diagnostyce i terapii,
- ☐ informatyczne systemy szpitalne,
- ☐ modelowanie matematyczne w medycynie,
- ☐ komputery w badaniach populacyjnych,
- ☐ analiza i modelowanie procesów rozwojowych,
- ☐ komputerowe wspomaganie dydaktyki w uczelniach medycznych.

Informacji udziela oraz przyjmuje zgłoszenia referatów i uczestnictwa w Konferencji:

Komitet Organizacyjny Konferencji Komputery w Medycynie
Instytut Informatyki Politechniki Łódzkiej
ul. Sterlinga 16/18, 90-217 Łódź, tel. 32-97-57.



SUPRA SP. Z O.O.

ul. Trybunalska 54, 60-325 POZNAŃ

telefon: 674392, telefaks: 236498

teleks: 412966

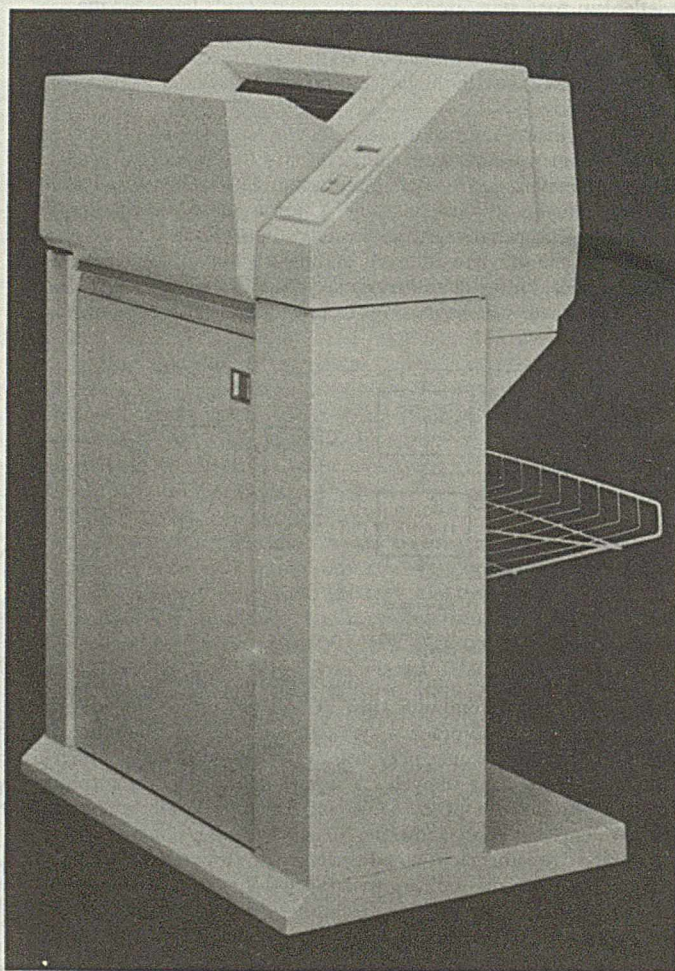
OFERUJE ŚWIATOWY BESTSELLER

**MANNESMANN
TALLY**

DRUKARKĘ MOZAIKOWO-WIERSZOWĄ MT 645

- doskonała jakość druku
- niezawodność 7000 h MTBF
- duża szybkość 450 wierszy/min
- cicha praca
- bufor 27 KB
- liczba kopii 1+5
- pełna grafika 240 (dpi)
- polskie litery (opcja)
- wewnętrzny program testujący
- dwa interfejsy automatycznie przełączane
- podwójny system traktorów

**MANNESMANN
TALLY**



NAJPRAWDOPODOBNIEJ NAJLEPSZE

DRUKARKI NA ŚWIECIE

AUTORYZOWANY DYSTRYBUTOR DRUKAREK

MANNESMAN TALLY



Struktura wieloprocessorowa do celów cyfrowego generowania obrazu

Znaczna złożoność obliczeniowa algorytmów cyfrowego generowania obrazu sprawia, że do ich realizacji w czasie rzeczywistym jest niezbędne zastosowanie specjalizowanych urządzeń. W pracy przedstawiono koncepcję architektury i funkcjonowania wieloprocessorowego systemu graficznego o strukturze potokowo-równoległej. Architektura ta łączy w sobie pozytywne cechy struktury potokowej oraz struktury równoległej, nie zawiera jednak ich wad. Opisano również niektóre szybkie przetwarzanie niezależnych bloków danych graficznych oraz czynniki mające wpływ na ostateczną postać architektury systemu. Atrakcyjną cechą opisanej architektury jest możliwość dokonywania istotnych modyfikacji bez utraty regularności. Przez zmianę liczby stopni oraz liczby jednostek w stopniach system może być konfigurowany stosownie do potrzeb konkretnego zastosowania.

Opisaną koncepcję architektury wykorzystano w projekcie toru generowania obrazu terenu, wchodzącego w skład systemu wizualizacji dla symulatora lotu.

Szybkie generowanie złożonych obrazów o dużej rozdzielczości wymaga mocy obliczeniowej przekraczającej możliwości pojedynczych układów o wielkim stopniu skalowania, w tym również specjalizowanych procesorów graficznych [5, 7]. Nowa generacja „superstacji” graficznych [1] wydaje się potwierdzać tę tendencję. Być może handlowa dostępność 32-bitowych, zmiennoprzecinkowych procesorów DSP [7, 9] oraz zaawansowanych procesorów uniwersalnych, jak 80486 lub i 860, spowoduje radykalną zmianę sytuacji.

W literaturze można znaleźć opisy kilku typowych rozwiązań stosowanych w stacjach graficznych i systemach symulacyjnych. Wspólną ich cechą jest oddzielenie sterowania, współpracy z systemem nadrzędnym itp., od przetwarzania danych graficznych. Funkcje sterownika są z reguły [2, 4, 6] realizowane przez procesor uniwersalny o bogatej liście rozkazów i regularnej architekturze (68000, 68010, 68020). Operacje graficzne są natomiast wykonywane przez specjalizowaną strukturę złożoną z wielu szybkich układów przetwarzających, jak np. procesory segmentowe serii AMD2900 [3] lub procesory sygnałowe, TMS32010 z wbudowanym równoległym układem mnożącym [4, 6]. Układy przetwarzające bywają łączone równoległe [10] lub potokowo.

Synchroniczna – w sensie logicznym – praca struktury potokowej oznacza w praktyce dopasowanie szybkości działania do najwolniejszego ze stopni, a struktura w pełni równoległa utrudnia równomierne obciążenie torów przetwarzających.

Architektura i podstawy działania systemu

Prezentowany system (tor) graficzny jest zbudowany z pewnej liczby stopni przetwarzających połączonych w ten sposób, że dane wejściowe każdego stopnia są wynikami przetwarzania realizowanego w stopniu poprzednim. Dane wejściowe dla pierwszego stopnia są dostarczane przez komputer nadrzędny; dane wyjściowe stopnia ostatniego są kierowane do sterownika pamięci obrazu.

Każdy stopień zawiera jedną lub więcej jednostek przetwarzających. Jednostki przetwarzające w całym torze działają współbieżnie i różnią się jedynie kodem wykonywanego programu. Wszystkie jednostki danego stopnia realizują asynchronicznie ten sam algorytm przetwarzania w odniesieniu do różnych porcji danych.

Przepływ danych między stopniami jest jednokierunkowy i buforo-

wany. Nie jest możliwe przesyłanie danych między jednostkami tego samego stopnia przetwarzającego. Buforowanie danych jest realizowane przy użyciu układów FIFO. Logiczną jednostką transmisji danych jest blok (np. opis wielokąta). W ogólnym przypadku wielkość i czas przetwarzania poszczególnych bloków mogą być różne, co wobec współbieżności działania jednostek w danym stopniu może prowadzić do zmiany kolejności bloków w strumieniu danych. W odróżnieniu jednak od innych zastosowań systemów wieloprocessorowych, dla wielu systemów graficznych zmiana kolejności przetwarzania porcji danych jest nieistotna, tzn. efekt przetwarzania danych zależy od samych danych. Przykładem może być obrazowanie trójwymiarowych scen z wykorzystaniem bufora głębokości (z-buffer): ostateczna postać obrazu nie zależy od kolejności operacji, a jedynie od odległości obiektów.

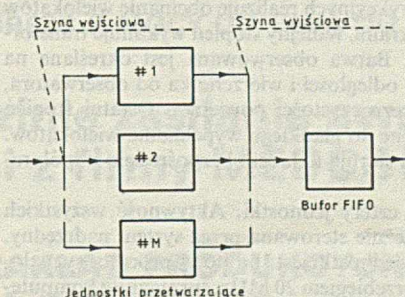


Rys. 1. Struktura toru graficznego

Oprócz kanału transmisji danych istnieje także w systemie kanał sterowania przetwarzaniem. Umożliwia on dwukierunkową komunikację między komputerem nadrzędnym a wybranym stopniem. Kanał ten jest wykorzystywany do przesyłania parametrów sterujących przetwarzaniem oraz dla potrzeb diagnostyki systemu.

Stopień przetwarzający zawiera połączone równoległe jednostki przetwarzające oraz wyjściowy bufor FIFO. Każdy z buforów współpracuje z jednostkami dwóch kolejnych stopni; jedne dokonują wyłącznie zapisu, a drugie – wyłącznie odczytu danych. Istotnymi elementami projektu toru graficznego są: podział algorytmu przetwarzania na etapy i ustalenie liczby jednostek w poszczególnych stopniach. Z uwagi na buforowanie danych oraz ograniczoną przepustowość dzielonych szyn danych, podział zadań między stopnie powinien być dokonywany z uwzględnieniem liczby i struktury wyników pośrednich generowanych w danym punkcie algorytmu.

Za optymalną należy uznać minimalną liczbę jednostek gwarantującą utrzymanie założonej przepustowości stopnia. Ponieważ czasowe aspekty przetwarzania mogą zależeć zarówno od algorytmów, jak i od danych, więc określenie a priori właściwej liczby jednostek wymaga przeprowadzenia symulacji działania stopnia na podstawie charakterystyk czasowych realizowanych w nim operacji oraz cech statystycznych strumienia danych.



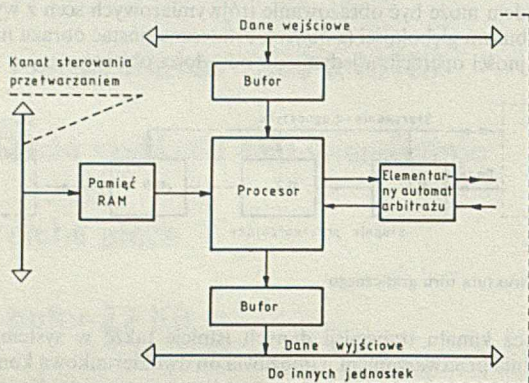
Rys. 2. Struktura stopnia przetwarzającego

Jest jednak możliwe dość proste oszacowanie maksymalnej liczby jednostek, przez określenie średniego udziału operacji związanych z przesyłaniem bloku danych w cyklu pracy jednostki przetwarzającej. Z wartości tej wynika, ile jednostek może w jednym cyklu kolejno współpracować z buforem FIFO. Znaczna rozbieżność wartości współczynników uzyskanych dla zapisu i odczytu danych świadczy o niepoprawnej dekompozycji algorytmu.

Jednostka przetwarzająca jest szybkim systemem mikroprocesorowym. Z dzielonymi buforami FIFO są związane dwie szyny: wejściowa i wyjściowa; szyna danych każdej z jednostek jest do nich przyłączona za pośrednictwem układów separujących. Zasadniczym warunkiem poprawnej pracy toru jest efektywny arbitraż dostępu jednostek do buforów FIFO (realizowany niezależnie dla bufora wejściowego i wyjściowego). Układ arbitrażu jest utworzony przez połączenie elementarnych automatów arbitrażu. Automat elementarny jest na stałe przypisany pewnemu procesorowi i stanowi integralną część jednostki przetwarzającej.

Synchronizacja

Istnieje kilka poziomów synchronizacji działania poszczególnych fragmentów toru graficznego. Synchronizacja na poziomie elementów jest zapewniana przez taktowanie wszystkich jednostek przetwarzających wspólnym przebiegiem zegarowym. Dla każdego stopnia istnieje lokalny mechanizm synchronizacji dostępu buforów FIFO. Problemy jednoczesności zapisu i odczytu danych do (z) bufora są rozstrzygane przez wewnętrzną logikę układów FIFO.



Rys. 3. Schemat blokowy jednostki przetwarzającej

Współpraca stopni przetwarzających jest synchronizowana przepływem danych. Zaburzenia logicznej struktury danych (podziału na bloki) powodują utratę synchronizacji międzystopniowej. Układy arbitrażu wykrywają możliwość wystąpienia takiej sytuacji i wymuszają odpowiednią reakcję procesora współpracującego w danym momencie z buforem.

Implementacja

Opisaną koncepcję architektury wykorzystano w projekcie toru generowania obrazu terenu wchodzącego w skład systemu wizualizacji dla symulatora lotu. Tor jest zbudowany z czterech stopni. Stopień transformacji geometrycznych wykonuje obliczenia związane z przekształcaniem współrzędnych wierzchołków wielokątów przy przejściu z układu odniesienia bazy danych do współrzędnych ekranowych. Macierz przekształcenia jest przesyłana na początku generowania każdej ramki obrazu.

Stopień transformacji restrykcyjnych realizuje obcinanie wielokątów zgodnie z liniami krawędzi ekranu. Kolejny stopień wykonuje transformację barwy wierzchołków. Barwa obserwowana jest określana na podstawie koloru wielokąta, odległości i wierzchołka od obserwatora, warunków oświetlenia i przezroczystości powietrza. Ostatni stopień wyznacza parametry potrzebne do gładkiego wypełniania wielokątów. Dane wyjściowe tego stopnia sterują układem bezpośrednio współpracującym z pamięcią obrazu.

Każdy ze stopni zawiera cztery jednostki. Aktywność wszystkich jednostek w torze jest niezależnie sterowana przez system nadrzędny. Elementem przetwarzającym jednostki jest 16-bitowy procesor sygnałowy TMS320C10 taktowany przebiegiem 20 MHz, sprzęgany z komputerem nadrzędnym przez wspólną pamięć. Kod programu jest wpisywany do lokalnej pamięci jednostki przed rozpoczęciem pracy systemu.

Dwuportowa pamięć lokalna ma pojemność $4 K \times 16$; jest możliwy jednoczesny zapis pamięci we wszystkich aktywnych jednostkach danego stopnia. Pojedynczy stopień zawiera 56 układów scalonych. Nominalna moc obliczeniowa toru wynosi 80 MIPS.

LITERATURA

- [1] Akeley K.: "The Silicon Graphics 4D/240GTX Superworkstation. IEEE Computer Graphics and Applications, pp. 71-84, July 1989
- [2] COLORIXGRAPH 90 Terminal Graphique Couleur. Materiały reklamowe, Traitement de l'Information Techniques Nouvelles, France
- [3] CUBI 7 System Graphique Couleur. Materiały reklamowe, TELMAT, France
- [4] Fujimoto A., Perrot C.G., Kansei Iwata: A 3-D Graphics Display System with Depth Buffer and Pipeline Processor. IEEE Computer Graphics and Applications, pp. 11-23, June 1984
- [5] Gutttag K.M. et al.: The TMS34010: An Embedded Microprocessor: IEEE Micro, pp. 39-52, June 1988
- [6] HS-50 Color Graphics Station. Materiały reklamowe, Genisco Computer Corp., USA
- [7] Papamichalis P., Simar R. Jr.: The TMS320C30 Floating-Point Digital Signal Processor. IEEE Micro, Vol. 8, No. 6, pp. 13-29, December 1988
- [8] Peterson R. et al.: Taking the Wraps off the 34020. Byte, pp. 257-272, September, 1988
- [9] Sohie G.R.L. Kloker K.L.: A Digital Signal Processor with IEEE Floating-Point Arithmetic. IEEE Micro, Vol. 8, No. 6, pp. 49-67, December 1988
- [10] Vital V Visual Simulation System. Materiały reklamowe, McDonnell Douglas Corp., USA.

Ogłoszenia można zgłaszać:

- | | |
|-----------------------|---|
| pisemnie | – 01-552 Warszawa, Pl. Inwalidów 10, p. 104 |
| telefonicznie | – nr 39-14-34 |
| telefaksem | – nr 20-31-16 |
| teleksem | – 814550 SIGMA PL |
| z zaznaczeniem | – Red. INFORMATYKI |



Przedsiębiorstwo Popierania Postępu TERA Spółka z o.o. uprzejmie informuje, że posiadając kilkunastoletnie doświadczenie w instalacji systemów wspomagających zarządzanie

o f e r u j e :

- opracowanie projektów systemów,
- optymalny kosztowo i rozwojowo dobór sprzętu i oprogramowania,
- instalację systemu u klienta,
- bezpłatnie przez rok konserwację oprogramowania oraz serwis sprzętu wraz z doradztwem techniczno-eksploatacyjnym,
- szybką dostawę uzupełnień konfiguracji lub sprzętu mikrokomputerowego niezależnego od systemu, w tym mikrokomputerów ALR firmy Wearnes Technology.

Wszelkie dodatkowe informacje uzyskają Państwo codziennie oprócz niedziel w Biurze Handlowym 40-025 Katowice, ul. 27 Stycznia 31/17 tel. (faks): 51-61-65, teleks: 315448 tera pl

PAMIĘTAJ! Instalacje XENIX/NOVELL/PC MOS 386 oraz serwis to nasza specjalność – ZAPRASZAMY, ponieważ czterech lat doświadczeń nigdzie nie kupisz.

01/19/90

MEDCOM

CHCESZ ZABEZPIECZYĆ BAZY DANYCH PRZED UTRATĄ INFORMACJI?

CHCESZ ZAPEWNIĆ PRAWIDŁOWĄ PRACĘ

KOMPUTERA * CENTRALI TELEFONICZNEJ * TELEFAKSÓW

W WARUNKACH ZANIKU NAPIĘCIA ZASILAJĄCEGO

kup ZASILACZE BEZPRZERWOWE:

naszej produkcji

UPS – 120 (abonenckie centrale telefoniczne i telefaksy)

UPS – 450

UPS – 850

UPS – 1450

produkcji tajwańskiej:

UPS – 500

Sprzedajemy również:

*** AKUMULATORY BEZOBSŁUGOWE, HERMETYCZNE FIRMY PANASONIC**

do zasilania urządzeń alarmowych, jedyne dopuszczone do używania w pomieszczeniach zamkniętych,

*** GENERATORY PRĄDOTWÓRCZE FIRMY ENDRESS O MOCY OD 1 kVA DO 350 kVA.**

**Twoje problemy w dziedzinie zasilania awaryjnego
rozwiążą specjaliści z firmy MEDCOM.**

BIURO

Al. Ujazdowskie 26/39

00-478 Warszawa

tel. 28-93-57, teleks 817060

ZAKŁAD PRODUKCYJNO-USŁUGOWY

ul. Bokerska 53

02-690 Warszawa

0/20/90

SDeL

język opisu scen geometrycznych

Jednym z ważnych problemów w komputerowym generowaniu obrazów jest możliwość łatwego tworzenia odpowiednich baz danych geometrycznych. Środowiska programowe i sprzętowe związane z definiowaniem takich baz danych bardzo silnie zależą od przewidywanego zastosowania i zawsze wymagają przyjęcia pewnego języka graficznego. Prezentowany język SDeL opracowano do tworzenia baz danych dla komputerowej symulacji lotu. Baza danych jest w tym zastosowaniu aproksymacją geometryczną pewnego wycinka terenu, uwzględniającą obiekty istotne dla wizualnej skuteczności wyświetlanych scen. Symulacja lotu polega natomiast na realizowaniu w czasie rzeczywistym „lotu nad bazą”.

Odzworowanie struktur geometrycznych w całym ich bogactwie na pewną notację sekwencyjną jest zagadnieniem nietrywialnym. Mnogość języków graficznych wynika z faktu, że różne typy obiektów geometrycznych i różne klasy relacji między tymi obiektami wymagają specjalnych mechanizmów językowych. Różne są również sposoby rozumienia (dekompozycji) scen geometrycznych. Ta sama scena płaska może być np. rozumiana jako nałożenie na płaszczyznę w ustalonej kolejności pewnej liczby „warstw” zawierających obiekty podstawowe (np. wielokąty wypukłe) albo jako rozbięcie płaszczyzny na rozłączne obszary (nie koniecznie wypukłe). Można wyobrazić sobie rysunek jako zbiór linii albo jako „ślad” pozostawiony przez urządzenie rysujące; w pierwszym przypadku opis rysunku jest specyfikacją zbioru linii, w drugim – sekwencją sterowań urządzenia rysującego, np. w konwencjach geometrii żółtowej języka Logo.

Język graficzny uważa się za język wysokiego poziomu, jeżeli występuje w nim rekurencyjne pojęcie obiektu złożonego, a więc obiektu zawierającego konkretyzację obiektów podstawowych i ewentualnie (innych) obiektów złożonych. Języki niskiego poziomu posługują się ustalonymi klasami obiektów podstawowych.

W językach graficznych najczęściej wykorzystuje się bezpośrednią specyfikę współrzędnych jako elementarny krok tworzenia opisu obiektu. Przykładami takich języków mogą być: DXF (ang. *Drawing Interchange Format*) w pakiecie graficznym AutoCAD lub język IPL (ang. *Intermediate Language for Pictures*) [1]. Niektóre języki, np. PostScript firmy ADOBE Systems [4], pozwalają jednolodnie manipulować tekstami, obiektami geometrycznymi i obrazami rastrowymi w ramach jednego dokumentu graficznego. Metoda bezpośredniej specyfikacji współrzędnych jest uniwersalna, ale w pewnych sytuacjach jej użycie może być utrudnione lub antyintuicyjne. Na przykład, zdefiniowanie trójkąta równobocznego wymaga obliczenia pierwiastka kwadratowego przy specyfikacji jednego z wierzchołków. W pracy [2] zaproponowano nową koncepcję specyfikacji obiektów, w której wykorzystano konstrukcję geometrii elementarnej jako operacje podstawowe przy tworzeniu opisu obiektu.

Przedstawiony język SDeL jest językiem wysokiego poziomu, w którym wykorzystuje się konwencję bezpośredniej specyfikacji współrzędnych. Ma on pewne pokrewieństwa pojęciowe z językiem DXF. Konstrukcje językowe podporządkowano głównemu zastosowaniu, jakim jest tworzenie baz danych dla komputerowej symulacji lotu. Baza danych jest w tym zastosowaniu aproksymacją geometryczną pewnego wycinka terenu, uwzględniającą obiekty terenowe istotne dla wizualnej autentyczności wyświetlanych scen.

Symulacja lotu polega na realizacji w czasie rzeczywistym „lotu nad bazą”. Zakładając, że w terenie nie występują obiekty ruchome, taka baza jest całkowicie statyczna. SDeL pozwala opisywać sceny bezpośrednio w postaci tekstowej i na tej podstawie tworzyć wewnętrzną

reprezentację bazy danych. Można również wykorzystywać notację SDeL jako postać pośrednią uzyskiwaną w wyniku konwersji obrazów stworzonych interakcyjnie za pomocą edytora graficznego.

Ogólny model sceny

Podstawowym wymaganiem w projekcie SDeL było stworzenie notacji pozwalającej opisywać wyobrażone przez użytkownika sceny terenowe. Mechanizmy języka powinny być zatem podporządkowane intuicyjnemu rozumieniu struktury tych scen i ułatwiać systematyczną konstrukcję opisu.

Przyjęty model koncepcyjny sceny jest następujący. Scena umieszczona w ustalonym prawoskrętnym układzie współrzędnych $OXYZ$ i jest ograniczona prostokątem o bokach równoległych odpowiednio do osi OX i OY . Prostokąt ograniczający dzieli się regularną siatką prostokątą na rozłączne sektory. Zawartość geometryczna każdego sektora tworzy się względem lokalnego układu współrzędnych, przy czym dowolną parę lub większą liczbę sektorów można związać z tym samym opisem. Istnieją trzy kategorie obiektów, które można rozmieszczać w sektorach: struktury wielokątowe dwuwymiarowe (2W), bryły i powłoki trójwymiarowe (3W) oraz światła stałe, rozblyskowe i kierunkowe. Każda z tych kategorii wymaga specyficznej reprezentacji i technik wizualizacji.

Struktura wielokątowa 2W składa się z warstw uporządkowanych wg priorytetów statycznych; obraz struktury płaskiej powstaje przez nałożenie na siebie obrazów kolejnych warstw.

Warstwa może zawierać hierarchię wielokątów wypukłych zdefiniowaną następująco: na najwyższym poziomie hierarchii jest określona lista wielokątów $w_1, w_2, \dots, w_i, \dots$, stanowiących główną dekompozycję warstwy; z każdym wielokątem w_i może być związana lista wielokątów niższego poziomu $w_{i1}, w_{i2}, \dots, w_{ij}, \dots$, itd., aż do uzyskania wymaganego stopnia szczegółowości. Nie narzuca się żadnych ograniczeń na głębokość dekompozycji ani nie ustala relacji geometrycznych, które powinny zachodzić między wielokątami w hierarchii (np. wymaganie, aby wielokąty tego samego poziomu były rozłączne).

W języku istnieją dwa podstawowe obiekty 3W: pryzmy i ostrosłupy. Ponadto można definiować dowolne bryły i powłoki przez specyfikację ścian. Ściana może być zdefiniowana początkowo w płaszczyźnie XY , a następnie umiejscowiona w przestrzeni trójwymiarowej za pomocą odpowiedniej transformacji. Ścianę nazywa się dekorowaną, jeśli w jej wnętrzu została zbudowana struktura hierarchiczna wielokątów; w tym przypadku wielokąty niższych poziomów hierarchii muszą być całkowicie zawarte w ścianie.

Światła są traktowane jako punkty w przestrzeni trójwymiarowej z dodatkowymi atrybutami określającymi barwę światła oraz właściwości czasowe (dla światła rozblyskowego) i parametry wiązki (dla światła kierunkowego).

Składnia i interpretacja języka

Ogólną strukturę opisu terenu podaje się w schemacie składniowym. Przyjęto standardowe symbole metajęzykowe $[]$, $\{ \}$ i $|$ do oznaczenia odpowiednio: wystąpienia opcjonalnego, interakcji i alternatywy. Dla wyróżnienia słowa kluczowe są pisane dużymi literami (ale język jest niewrażliwy na wielkość czcionki), kursywą zaś zaznaczono identyfikatory lub nazwy odpowiednich pojęć składniowych. Pozostałe symbole oznaczają siebie.

Opis terenu składa się z trzech głównych sekcji: definicji obiektów, definicji bloków i definicji sektorów. W nagłówku opisu podaje się nazwę terenu, stałe całkowite określające rozmiar siatki sektorów i opcjonalnie długości boków sektora. Definicje w sekcji DEFINE opisu

TERRAIN nazwa_terenu (m,n); [SECTSIZE size_x,size_y;]	opis_terenu
<pre>[DEFINE [nazwa = POINT (x,y,[z]); nazwa = LINE (p/p/p); nazwa = POLYGON(p/p/p); nazwa = PRISM wielokąt TOP transformacja; nazwa = PYRAMID wielokąt APEX punkt; nazwa = PLIGHT (kolor); nazwa = VLIGHT (kolor,rozbiysk); nazwa = DLIGHT (kolor,kierunek); nazwa = COLOR (kolor); nazwa = FLASH (rozbiysk); nazwa = FLUX (kierunek); nazwa = RGB (r,g,b); nazwa = {stała,{stała}}; nazwa = stała;]</pre>	definicje_obiektów
<pre>{BLOCK nazwa_bloku [(par{,par})]; definicje_obiektów</pre>	definicje_bloków
<pre>{PUT [(k/dp)] obiekt_podstawowy [(t/s/r)] [WITH atrybuty]; INSERT [(k/dp)] nazwa_bloku([arg{,arg}]) [(t/s/r)]; SOLID [FACE struktura_płaska [WITH atrybuty];] ENDSOLID; SHELL [FACE struktura_płaska [WITH atrybuty];] ENDSHELL; ENDBLOCK; }</pre>	specyfikacja_bloku
<pre>{ SECTOR snazwa(i,j{[i,j]}) EXTERN; SECTOR snazwa(i,j{[i,j]}) definicje_obiektów_lokalne_w_sektorze definicje_bloków_lokalne_w_sektorze [CLUSTER nazwa_agregacji(priorytet); definicje_obiektów_lokalne_w_agregacji specyfikacja_bloku ENDCLUSTER;] ENDSECTOR; }</pre>	definicje_sektorów
<pre>[START specyfikacja_położenia_obserwatora;] [PALETTE specyfikacja_pliku;] END</pre>	

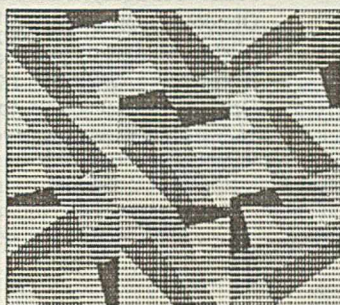
Przykłady

```
TERRAIN pole(1,1);

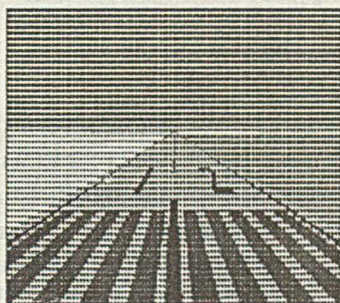
SECTOR sekt1(1,1);
DEFINE
w1=polygon(0,0/0,1/2,2/3,0);
w2=polygon(2,2/5,3/7,0/3,0);
w3=polygon(5,3/8,4/9,0/7,0);
w4=polygon(8,4/10,5/10,0/9,0);
w5=polygon(0,1/0,3/7,7/8,4);
w6=polygon(7,7/10,10/10,5/8,4);
w7=polygon(0,5/6,10/7,7/0,3);
w8=polygon(6,10/10,10/7,7);
w9=polygon(0,10/6,10/0,5);
```

```
BLOCK b1();
PUT w1 WITH green;
PUT w2 WITH brown;
PUT w3 WITH red;
PUT w4 WITH yellow;
PUT w5 WITH lgray;
PUT w6 WITH lblue;
PUT w7 WITH lgreen;
PUT w8 WITH magenta;
PUT w9 WITH cyan;
ENDBLOCK
```

```
CLUSTER agregat(0);
insert b1() (0,0 /3,3/);
insert b1() (60,0/3,3/90);
insert b1() (90,30/3,3/180);
insert b1() (0,60/3,3/-90);
insert b1() (60,30/3,3/90);
insert b1() (60,60/3,3/-90);
insert b1() (30,60/3,3/90);
insert b1() (30,90/3,3/-90);
insert b1() (60,60/3,3/);
ENDCLUSTER;
ENDSECTOR;
END
```



Sektor "sekt1", widok z góry



Widok pasa startowego

terenu mają zasięg globalny. Można nadać dowolnemu obiektowi podstawowemu (POINT, LINE, POLYGON, PRISM, PYRAMID, PLIGHT, VLIGHT, DLIGHT), a także atrybutom koloru (COLOR), barwy (RGB), rozbiysku (FLASH) i wiązki światła (FLUX). Ponadto można zdefiniować wektor wartości lub wartość skalarną. Wszędzie dalej w opisie, gdzie jest spodziewany pewien obiekt podstawowy, atrybut, wektor lub skalar, można użyć odpowiedniej nazwy ze skutkiem równoważnym substytucji.

Definiowanie bloków jest najważniejszym mechanizmem strukturalizacji opisu. Blok jest parametrycznym obiektem złożonym, w którym mogą wystąpić obiekty podstawowe, bryły, powłoki i konkretyzacje innych (wcześniej zdefiniowanych) bloków. Wewnątrz boku można użyć sekcji definicji obiektów o zasięgu lokalnym z przysłanianiem homonimów w definicjach globalnych.

Operacja PUT służy do umieszczania w bloku obiektów podstawowych jawnych albo nazwanych w sekcji DEFINE (lokalnej lub globalnej). Operacja ta pozwala przed wstawieniem przekształcić obiekt za pomocą transformacji liniowej ($t/s/r$), przy czym t , s , r , oznaczają odpowiednio przesunięcie, skalowanie i obroty. Wstawienie może być pojedyncze lub grupowe; w tym drugim przypadku specyfikacja grupy (K/dp) określa krotność wstawienia k przesunięcia między parą kolejnych obiektów w grupie. Obiektowi można nadać atrybuty, w szczególności kolor. Operacja INSERT dotyczy wstawiania bloków i jest podobna do operacji PUT; atrybuty wstawienia podaje się za pomocą ogólniejszego mechanizmu parametrów bloku.

Wstawienie dowolnej bryły lub powłoki realizują konstrukcje SOLID i SHELL (bryły standardowe PRISM i PYRAMID są traktowane tak samo, jak inne obiekty podstawowe). Składniowo konstrukcje te są równoważne; różnica semantyczna polega na odmiennych warunkach topologicznych w odniesieniu do brył i powłok. Specyfikacja ściany (FACE) odwołuje się do struktury płaskiej, którą może być wielokąt jawny, nazwa wielokąta albo odsyłacz do nazwy bloku. W przypadku odwołania do bloku ścianę stanowi tylko struktura wielokątowa płaska w specyfikacji bloku. Jeżeli blok zawiera hierarchię wielokątów, to za ścianę przyjmuje się pierwszy wielokąt najwyższego poziomu wraz z podległymi mu wielokątami niższych poziomów. Jest to ściana dekorowana

Sektory definiuje się jako samoistne (pod)sceny. W nagłówku oprócz nazwy jest podana lista indeksów sektorów, które mają uzyskać identyczną zawartość geometryczną pochodzącą z biblioteki (EXTERN) albo z opisu jawnego. W opisie sektora można zdefiniować obiekty i bloki lokalne wg tych samych zasad, jak na poziomie opisu terenu. Treść sektora składa się z ciągu agregacji (skupienia) obiektów (konstrukcja CLUSTER) o budowie różniacej się tylko nagłówkiem od struktury bloku. Różnica w interpretacji bloku i agregacji polega na tym, że blok jest definicją parametryczną pewnego obiektu złożonego, agregacja zaś jest jednorazową konkretyzacją obiektu złożonego. Struktury wielokątowe płaskie w skupieniach CLUSTER są traktowane jako warstwy z priorytetem podanym w nagłówku.

Opis terenu może być rozmieszczony w jednym lub większej liczbie plików tekstowych. Plik, w którym znajduje się początek i koniec opisu jest plikiem głównym. Pliki pomocnicze włącza się dyrektywą `#include <plik>`, rozpoznawaną na poziomie analizy leksykalnej.

LITERATURA

- [1] ten HAGEN P.J.W. et al.: ILP- Intermediate Language for Pictures. Math. Center Tracts, No. 130, pp. 110, Amsterdam 1980
- [2] NOMA T., KUNII T.L. et al.: Drawing Input Through Geometric Constructions: Specification and Applications. New Trends in Computer Graphics, pp. 403-415, Springer-Verlag 1988
- [3] Pająk A.: A Simple Descriptive Language for Definition of Geometric Scenes. Instytut Informatyki pp. 15, Warszawa 1989
- [4] PostScript Language Reference Manual. Addison-Wesley, Reading 1985.

Z przykrością informujemy Zleceniodawców ogłoszeń zamieszczanych w INFORMATYCE o konieczności podwyższenia ich ceny począwszy od lutego br., proporcjonalnie do wzrostu kosztów druku i papieru oraz ceny czasopisma.

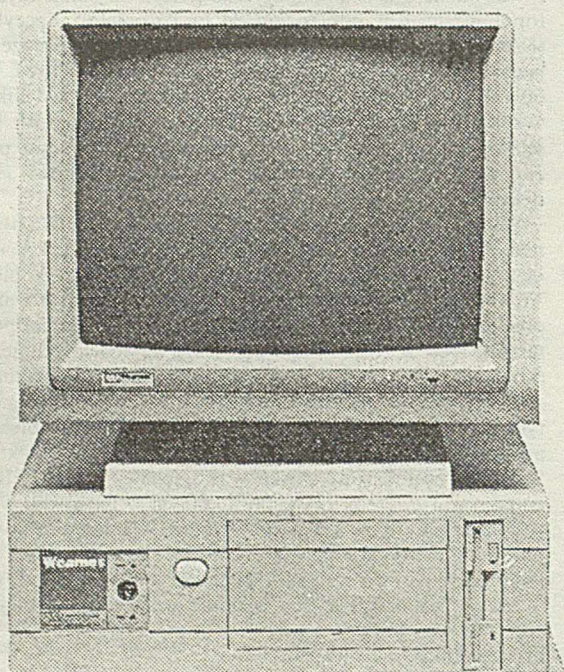
O nowych cenach Zleceniodawcy zostaną indywidualnie powiadomieni.

KOMPUTER Z WBUDOWANĄ PRZYSZŁOŚCIĄ

Komputer zaprojektowany przez czołowego światowego producenta mikrokomputerów – firmę ADVANCED LOGIC RESEARCH i wyprodukowany przez firmę WEARNES TECHNOLOGY.

Komputer, którego możliwości i cena oszałamiają konkurencję.

Komputer, który nie zestarzeje się, dzięki możliwości rozbudowy do architektury 386SX i 486.



Dane techniczne:

SERIA WEARNES BOLDLINE „M”

- Procesor 80286 – 12.5 Mhz
- Pamięć 1 MB RAM, możliwość rozbudowy do 16 MB (5 MB na płycie głównej)
- BIOS Phoenix
- Napęd dyskietek 3,5" 1,44 MB
- Zasilacz 110 Watt
- Klawiatura 101 klawiszy
- Podstawka dla koprocatora matematycznego 80287
- Wbudowany sterownik dyskowy z przepływem 1:1
- Port szeregowy i równoległy
- 40 MB dysk sztywny
- Możliwość korzystania z EMS 4.0
- Obudowa typu „compact”
- Miejsce na dwa napędy 5,25" o wysokości 1/2
- Podręcznik i dyskietka z programem konfiguracyjnym
- Opcjonalna rozbudowa do 386SX i 486
- 12-miesięczna gwarancja

BOLDLINE COMPUTERS
GRUPA MICOMP-TECH
Biuro Informacji Techniczno-Handlowej
ul. Astrów 7, 40-045 Katowice
telefon i telefaks: 518-628
teleks: 315687 COMP PL

DYSTRYBUTORZY:
PTH „TECHMEX”
43-300 Bielsko-Biała
ul. M. Curie Skłodowskiej 13
tel.: 42-198, 47-555, telefaks: 47-624,
teleks: 35325

Układ szybkiego wypełniania pamięci obrazu

Wypełnianie pamięci obrazu w rastrowym systemie graficznym może być realizowane wieloma metodami, zależnie od struktury systemu, uwarunkowań czasowych, specyfiki generowanych obrazów itp. W opracowaniu przedstawiono koncepcję i opis implementacji autonomicznego sterownika VRAM do szybkiego (z użyciem algorytmu Gourauda) wypełniania pamięci obrazu. Sposób ten zapewnia gładkie cieniowanie powierzchni wyświetlanych wielokątów. Opisano także rozwiązania zastosowane w celu zwiększenia szybkości działania układu. Przedstawiona koncepcja została opracowana na potrzeby systemu wizualizacji dla symulatora lotu.

Sterowanie pamięci obrazu w systemach czasu rzeczywistego wymaga zastosowania efektywnych algorytmów i szybkich układów [3, 6, 7]. Na przykład, generowanie kadrów obrazu o wielkości 1024×768 j.w. z częstotliwością 60 Hz oznacza konieczność zapisu kolejnych pikseli, średnio co ok. 20 ns. Odpowiada to w przybliżeniu czasowi propagacji sygnału przez trzy bramki FAST – najszybszej dostępnej komercyjnie serii układów TTL. Ponadto sprzętowy charakter pewnych metod generowania obrazu sprawia, że wartości tych samych pikseli są zapisywane wielokrotnie – oczywiście powoduje to dodatkowe zastrzeżenie wymagań czasowych. Oszacowany w ten sposób wymagany czas generowania piksela wynosi 5...10 ns, co w zestawieniu z typowym czasem trwania cyklu instrukcyjnego procesorów stosowanych w stacjach graficznych [1, 3, 5] (ok. 200 ns), ukazuje istotę problemu.

Funkcje układu

Układ steruje wypełnianiem pamięci w systemie, w którym obrazowane sceny są tworzone z barwnych wielokątów. Początkowe kroki przetwarzania wiążą się z transformacjami współrzędnych, obcinaniem i wyznaczaniem obserwowanej barwy wierzchołków. Zgodnie z algorytmem Gourauda [4] barwa punktów krawędzi wielokąta jest obliczana jako kombinacja liniowa kolorów przypisanych wierzchołkom leżącym na tej krawędzi. Wielokąt jest następnie rozkładany na odcinki wyznaczone przez punkty przecięcia krawędzi wielokąta z kolejnymi liniami rastra obrazu. Opisy odcinków są przesyłane do sterownika, gdzie podlegają rozwijaniu.

Algorytm generowania odcinka

Znajomość długości odcinka N oraz koloru jego piksela początkowego C_p i piksela końcowego C_k , pozwala obliczyć jednostkowy przyrost koloru dC niezbędny do uzyskania gładkiego wypełnienia odcinka:

$$dC = (C_k - C_p) / N$$

Barwa i -tego piksela może być wówczas wyznaczona jako

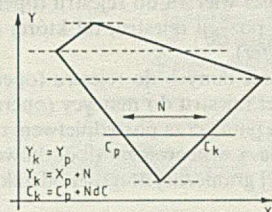
$$C_i = C_p + i dC \quad i = 0, \dots, N-1$$

lub w postaci reguły iteracyjnej:

$$C_{i+1} = C_i + dC \quad C_0 = C_p, \quad i = 0, \dots, N-2$$

Formuła ta stanowi podstawę działania prezentowanego sterownika. Generowanie odcinka sprowadza się do wykonania $N-1$ sumowań z akumulacją wyniku, przy czym do pamięci jest przesyłana wartość

początkowa C_p i wszystkie sumy. Przedstawione operacje są realizowane niezależnie w odniesieniu do składowych RGB barwy.



Rys. 1. Znaczenie parametrów opisu odcinka

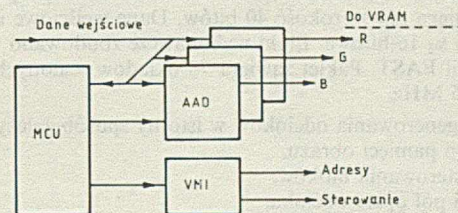
Blok opisu odcinka zawiera pola określające:

- długość odcinka N ,
- współrzędne ekranowe piksela początkowego $[X_p, Y_p]$,
- kolor piksela początkowego $C_p = [R_p, G_p, B_p]$,
- przyrost koloru $dC = [dR, dG, dB]$.

Postać realizowanej formuły sprawia, że przy niedostatecznej dokładności obliczeń, w kolejnych krokach algorytmu występuje kumulacja błędów. Ponadto w odróżnieniu od składowych koloru C_p opisywanych nieujemnymi liczbami całkowitymi, składowe przyrosty dC mogą przyjmować wartości ujemne i (lub) ułamkowe. Wymagana jest zatem różna reprezentacja poszczególnych pól w bloku opisu. Z racji przyrostowego charakteru operacji arytmetycznych, układ wypełniający będzie dalej nazywany inkrementerem.

Architektura

Na całość modułu inkrementera składają się: jednostka sterująca MCU, układ sprzęgający pamięci VMI i sumatory akumulujące AAD (rys. 2). Dane wejściowe są dostarczane przez układy przetwarzające opisy wielokątów; wyjścia inkrementera sterują pamięcią. Realizacja pamięci obrazu z układów VRAM wydaje się rozwiązaniem naturalnym wobec atrakcyjnych cech architektury i zwiększającej się pojemności tych układów.



Rys. 2. Schemat blokowy układu

Wbudowany rejestr przesuwany umożliwia wielokrotne zmniejszenie narzutów czasowych związanych z wyświetlaniem zawartości pamięci VRAM. Zadaniem sumatorów akumulujących jest wyznaczanie wartości składowych barwy kolejnych pikseli. Precyzja realizowanych operacji wynika z długości reprezentacji składowych przyrostu dC . Układ sprzęgający pamięci wytwarza sygnały niezbędne do współpracy z pamięcią obrazu. Działanie generatora sygnałów sterujących zależy głównie od typu użytych układów VRAM; metoda generowania adresów zależy ponadto od fizycznej organizacji pamięci.

Sterowanie pamięci obrazu

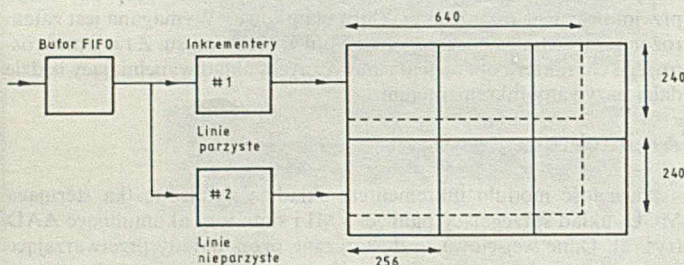
Jeśli wypełnianie oraz wyświetlanie dotyczy jednego zestawu pamięci, to zapis danych jest dokonywany od strony sprzęgu równoległego, a odczyt kolejnych wierszy matrycy – za pośrednictwem rejestru przesuwanego. Ponieważ z zasady działania inkrementera wynika, że zapisywane piksele należą do tej samej linii rastra, można zastosować tzw. stronicowy tryb dostępu (ang. *page mode*). Adres wiersza jest wówczas przesyłany tylko raz dla całego ciągu danych.

W wielu systemach stosuje się buforowanie wyświetlanego obrazu; zapis i odczyt dotyczą wtedy oddzielnych kompletów pamięci (zasada *ping-pong*). Przelaczanie kompletów następuje po zakończeniu generowania nowego kadru obrazu. Struktura taka umożliwia wykorzystanie rejestru przesuwanego do celów wypełniania pamięci. Pozwala to zwiększyć szybkość zapisu w porównaniu z dostępem stronicowym. Ponieważ jednak przesyłanie między rejestrem a matrycą dotyczących wierszy, przy generowaniu odcinka trzeba wykonać następujące operacje:

- odczytanie zawartości wiersza do rejestru (operacja *read transfer*),
 - ustawienie numeru pozycji rejestru, od której rozpocznie się zapis (operacja *pseudo transfer*),
 - sekwencyjne zapisanie danych do rejestru (operacja *serial write*),
 - zapisanie zawartości rejestru do matrycy (operacja *write transfer*).
- Jak widać, wypełnianie pamięci za pośrednictwem rejestru przesuwanego komplikuje sterowanie i wprowadza dodatkowe narzuty czasowe. Jednak powyżej pewnej granicznej długości odcinka ta metoda generowania jest szybsza.

Implementacja

Przedstawiona koncepcja została wykorzystana w projekcie systemu wizualizacji dla symulatora lotu. Obraz o rozmiarach 640×480 j.v. jest generowany z częstotliwością 30 Hz a kolor piksela jest zapisywany w 15 bitach. Pracująca według metody ping-pong pamięć obrazu została zbudowana z układów VRAM $64 K \times 4$. Pojedynczy komplet pamięci jest tworzony przez dwa banki. W jednym odwzorowane są parzyste, a w drugim – nieparzyste linie obrazu. Każdy z banków ($192 K \times 16$) współpracuje z oddzielnym modulem inkrementera. Opisy odcinków są pobierane ze wspólnego bufora FIFO.



Rys. 3. Organizacja i sterowanie pamięci obrazu

Sterowanie jest realizowane przez strukturę mikroprogramowaną; pamięć sterująca ma szerokość 40 bitów. Dane wejściowe i sumatory akumulujące są 16-bitowe. Bloki wykonawcze zbudowano z układów MSI/SSI serii FAST. Pakiet zawiera 75 układów scalonych i pracuje z zegarem 25 MHz.

Szybkość generowania odcinków w istotny sposób zależy od:

- organizacji pamięci obrazu,
- sposobu sterowania bloków,
- kolejności pól opisu odcinka,
- przepływu sterowania w mikroprogramie.

Pamięć obrazu jest zorganizowana w sposób umożliwiający niezależne adresowanie bloków przechowujących różne segmenty linii rastra (rys. 3). Cykle przesłania są wykonywane jednocześnie we wszystkich blokach, a adres w cyklach „*pseudo transfer*” wynosi $X_3 \bmod 256$ w bloku zawierającym piksel początkowy odcinka, oraz 0 w pozostałych blokach. Niezależne, tzw. poziome sterowanie układów wykonawczych oraz odpowiednia postać danych wejściowych umożliwiają rozpoczęcie przetwarzania już podczas wczytywania bloku opisu. Szybka realizacja rozejść (rozwidleń) warunkowych mikroprogramu jest ułatwiona dzięki zastosowaniu dwuadresowego układu szeregującego. Dalsze zwiększenie szybkości wypełniania może zostać uzyskane przez podział obrazu na wiele sfer i zwielokrotnienie modułu inkrementera. Pożądane jest

wówczas zastąpienie dzielonego bufora FIFO specjalizowanym blokiem sortującym opisy według stref.

★ ★ ★


Przedstawiona metoda generowania odcinka może zostać uogólniona dla przypadków trapezu wyznaczonego przez krawędzie wielokąta i linie rastra zawierające wierzchołki (rys. 1). Przy odpowiedniej postaci bloku opisu jest możliwe przyrostowe generowanie współrzędnych oraz barwy pikseli brzegowych, a na ich podstawie także pikseli wnętrza wielokąta. Następnym etapem rozbudowy funkcji układu wypełniającego jest autonomiczne przetwarzanie całych wielokątów oraz wyznaczanie wartości współrzędnej Z, wymaganych w systemach z buforem głębokości [7].

Innym interesującym problemem jest generowanie odcinków, połączone z wygładzaniem krawędzi (ang. *anti-aliasing*). Konieczne jest w takim wypadku przeliczanie wartości pikseli brzegowych z uwzględnieniem koloru tła oraz korzystanie ze współrzędnych subpikselowych [2].

LITERATURA

- [1] CUBI 7 System Graphique Couleur. Materiały reklamowe, TELMAT, France
- [2] Fujimoto A., Iwata K.: Jag-Free Images on Raster Displays. IEEE Computer Graphics and Applications, pp. 26–34, December 1983
- [3] Fujimoto A., Perrot C.G., Iwata K.: A 3-D Graphics Display System with Depth Buffer and Pipeline Processor. IEEE Computer Graphics and Applications, pp. 11–23, June 1984
- [4] Gouraud H.: Continuous Shading of Curved Surfaces. IEEE Transactions on Computer, Vol. C-20, No. 6, pp. 623–629, June 1971
- [5] HS-50 Color Graphics Station. Materiały reklamowe, Genisco Computer Corp., USA
- [6] SL1001/SL1002 Graphics Microprocessor Chip Set. Materiały reklamowe, SubLOGIC Corp., USA
- [7] Swanson R.W., Thayer L.J.: A Fast Shaded-Polygon Renderer. SIGGRAPH 86, Vol. 20, No. 4, pp. 95–100, 1986.

Tylko prenumerata gwarantuje regularne otrzymywanie INFORMATYKI



**PROMOCYJNA
80% OBNIŻKA CEN!**

LOGOTEC Engineering

Oferujemy

licencjonowane, polskie wersje językowe zachodniemieckich systemów wspomagania projektowania

LOGOCAD – mechanika ELOCAD – elektrotechnika

oprogramowanie pomocnicze

pakiety ponad 100 polskich (PN) i niemieckich (DIN) norm mechanicznych (śruby, nakrętki, podkładki, pierścienie, zawlecзки, sworznie, sprężyny, łożyska toczne itp.)

inżynierskie programy obliczeniowe:

(wały, belki, ramy, kraty, ruszty, płyty, powłoki, płaski stan naprężeń i płaski stan odkształceń, przekroje cienko- i pełnościenne, obliczenia i optymalizacja kół zębatych)

programy do obsługi baz danych

UNIP2 – uniwersalny generator wydawnictw i programów standardu dBase i Clipper

nagrodzony MikroLaurem '88 i '90

ARCHIW – archiwum rysunków i projektów

MOL (książkowy) – do kompleksowej obsługi biblioteki i wypożyczalni książkowej

doradztwo i szkolenia

z zakresu zastosowania naszych systemów CAD, kursy obsługi naszego oprogramowania, pomoc we wdrożeniu i eksploatacji programów

sprzęt

kompletne stanowiska dla naszych systemów, digitizery, monitory i karty graficzne dla różnych systemów CAD.

Jesteśmy do państwa dyspozycji

LOGOTEC ENGINEERING

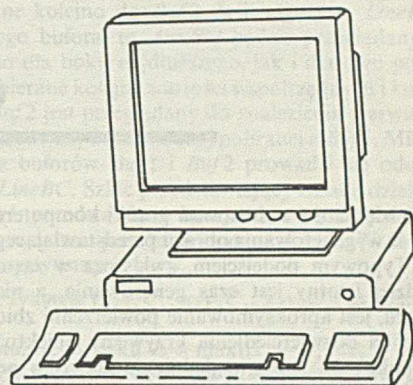
Zakład Rozwoju Systemów CAD

Nasz nowy adres: 40-028 KATOWICE, ul. Francuska 70
tel.: 57-46-26, 57-23-76, 51-17-47, teleks 0312606

0/21/90



POLSKA SPÓŁKA INŻYNIERSKA
Sp. z o.o.



TERMINAL AVT 180-PC

Szybki terminal znakowy AVT 180-PC przeznaczony jest do pracy w systemach wielostanowiskowych opartych na mikrokomputerach klasy IBM PC XT/AT/386, MICROVAX, PDP-11 i innych. Spotykane na rynku polskim konstrukcje zachodnie przewyższa następującymi parametrami:

- Realizuje specjalne protokoły dla systemów QNX, PC-MOS, MULTI-LINK. Dzięki temu praca na terminalu nie różni się od pracy na konsoli głównej.
- Parametry obrazu są identyczne jak w IBM PC z kartą Hercules.
- Terminal jest wyposażony w 18 generatorów znaków (9 polskich, Hercules, 2 ANSI, 5 rosyjskich i czeski). Na ekranie można jednocześnie wyświetlać wszystkie znaki z różnych generatorów co – na przykład – pozwala na napisanie części tekstu po rosyjsku, a resztę po polsku.
- Terminal pozwala na szeroki wybór układu znaków narodowych na klawiaturze. Dostępnych jest 10 układów polskich, 6 rosyjskich, amerykański, angielski, francuski, niemiecki i hiszpański.

Terminal może pracować w następujących protokołach:

PCSHADOW – pod kontrolą systemu **MULTILINK**.

PCTERM – pod kontrolą systemu **PC-MOS**.

QNXS – pod kontrolą systemu **QNX**.

VT52+ – pod kontrolą systemu **XENIX/UNIX** i innych. Protokół rozwinięty o atrybuty znaku i sekwencje klawiatury.

ANSI 3.64 – pod kontrolą systemu **XENIX/UNIX** i innych. Protokół zgodny z normą ANSI 3,64 i ISO 6429.

VT 100 – pod kontrolą systemu **RSX 11** i innych. Protokół ograniczony do 80 znaków w linii.

VT 220 – zaimplementowany jest podzbiór protokołu VT 220.

Adres: 02-567 Warszawa, ul. Sandomierska 13
Telefon: 49-56-74, 48-97-73, Teleks: 817795 dglb pl

ZO/317/90

Algorytm rysowania barwnego odcinka i jego wykorzystanie do cieniowania wnętrza trójkąta

W większości zastosowań grafiki komputerowej pojawia się konieczność wygenerowania obrazu przedstawiającego obiekty trójwymiarowe. Typowym podejściem, zwłaszcza w zastosowaniach, w których bardziej istotny jest czas generowania, a nie fotograficzny realizm obrazu, jest aproksymowanie powierzchni zbiorem wielokątów.

Próba odzwierciedlenia krzywizny obiektu oświetlonego światłem kierunkowym lub uwzględnienie zjawiska perspektywy powietrznej w obrazie terenu dla różnego rodzaju symulatorów prowadzi do wyświetlania wielokątów, których wierzchołki mogą przyjmować barwę i jasność różną od oryginalnej barwy i jasności obiektu [3]. W fazie generowania jest konieczne wygenerowanie barwy i jasności pikseli leżących wewnątrz wielokąta, przy uwzględnieniu faktu, że każdy z wierzchołków może mieć różną barwę i jasność. Tego rodzaju wypełnianie wnętrza figury będzie nazywane cieniowaniem wnętrza, dla odróżnienia od wypełniania jednolitą barwą. Pojęcie cieniowania dla zastosowań monochromatycznych zostało wprowadzone przez Gourauda [2]. Ze względu na to, że kształt dowolnej powierzchni można aproksymować trójkątami, rozważania ograniczono do cieniowania wnętrza trójkąta. Pewnym ułatwieniem jest fakt, że trójkąt jest zawsze wielokątem wypukłym. Poniżej przedstawiono algorytm barwnego cieniowania wnętrza trójkąta o zadanych barwach wierzchołków. Do cieniowania wnętrza równoległymi odcinkami wykorzystuje się algorytm kreślenia odcinka o zadanych barwach punktu początkowego i końcowego.

Algorytm rysowania barwnego odcinka

Pełny opis piksla wyświetlanego na ekranie barwnego monitora graficznego wymaga podania współrzędnych ekranowych x_e, y_e oraz informacji o kolorze piksla C . W zależności od architektury monitora w pamięci obrazu może być przechowywana trójka liczb reprezentujących wartości składowych trójkolorowych r', g', b' albo jedna liczba stanowiąca wskaźnik do tabeli kolorów n . W pierwszym przypadku opis piksla ma postać:

$$p = (x_e, y_e, r', g', b')$$

$$x_e \in \langle 0, HRes \rangle, y_e \in \langle 0, VRes \rangle, r', g', b' \in \langle 0, 2^k - 1 \rangle$$

a w drugim:

$$p = (x_e, y_e, n) \quad n \in \langle 0, 2^l - 1 \rangle$$

gdzie: $HRes, VRes$ – wielkość pozioma i pionowa ekranu w jednostkach rastra,

k – liczba bitów słowa reprezentującego składowe r', g', b' ,

l – liczba bitów wskaźnika do tabeli kolorów.

Dla uproszczenia zapisu można przyjąć, że barwa i jasność piksla stanowią jego kolor; kolor ten będzie traktowany jako wektor składowych trójkolorowych $C[r', g', b']$. Jeśli jest określona tabela kolorów, to kolor piksla jest reprezentowany przez wektor jednoelementowy $C[n]$. Pełny opis koloru w postaci składowych r', g', b' jest wówczas zawarty w tabeli kolorów w komórce o adresie n .

Wyświetlenie na ekranie odcinka $x_p, y_p, c_p - x_k, y_k, c_k$ wymaga wyznaczenia współrzędnych ekranowych x_{ei}, y_{ei} kolejnych pikseli odcinka oraz określenia ich koloru C_i przez dokonanie interpretacji między kolorem C_p a kolorem C_k dla każdego piksla

$$x_{ei}, y_{ei},$$

$$\begin{aligned} C_i &= C_p \cdot [1 - i/(N-1)] + C_k \cdot i/(N-1) & i \in \langle 0, N-1 \rangle \\ C_i &= C_p + (C_k - C_p) \cdot i/(N-1) & \text{gdzie } N - \text{liczba pikseli odcinka} \\ C_i &= C_p + dC \cdot i & dC = (C_k - C_p)/(N-1) \end{aligned}$$

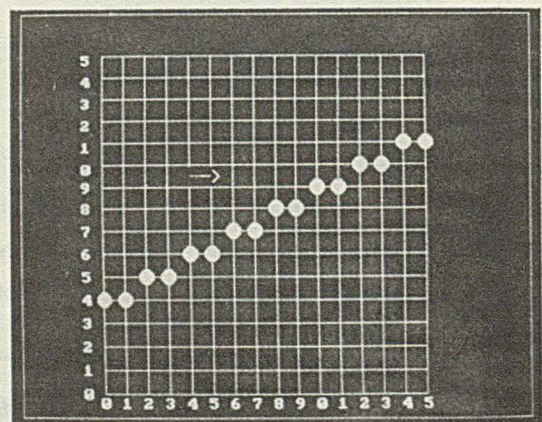
lub w postaci iteracyjnej:

$$C_0 = C_p; \quad C_i = C_{i-1} + dC \quad i \in \langle 1, N-1 \rangle$$

Powyższe formuły są wykorzystywane w algorytmach klasy DDA [3] wymagających stosowania reprezentacji liczb rzeczywistych dla zmiennej dC oraz wykonania operacji dzielenia.

Poniżej przedstawiono algorytm *LineBC*, będący rozszerzeniem koncepcji algorytmu *Bresenhama*. W oryginalnym algorytmie [1] jest wyróżniona zmienna niezależna (o większej różnicy współrzędnych punktu końcowego i początkowego), zwiększana każdorazowo, oraz zmienna zależna zmieniana warunkowo, w zależności od wartości zmiennej decyzyjnej *Eps*. W dwu wymiarach na płaszczyźnie wyczerpuje to zestaw zmiennych. Zachowując to podejście można wyróżnić zmienną niezależną w dowolnej liczbie wymiarów. Wartości składowych r', g', b' , albo wartość wskaźnika tabeli kolorów n są traktowane na równych prawach ze współrzędnymi ekranowymi x_e, y_e , stanowiąc współrzędne piksla w przestrzeni pięcio- albo trójwymiarowej. W przypadku występowania równych różnic, za zmienną niezależną należy przyjmować współrzędną ekranową. W przypadku równości współrzędnych ekranowych, za zmienną niezależną przyjmuje się współrzędną x_e . Po wyróżnieniu zmiennej niezależnej wszystkie pozostałe stają się zmiennymi zależnymi. Dla każdej z nich jest wyznaczona odrębna zmienna decyzyjna *Eps*. W kolejnych krokach algorytmu wykonuje się zwiększanie współrzędnej niezależnej oraz rozejścia na krok poosiowy albo diagonalny dla każdej ze zmiennych zależnych, na podstawie wartości odpowiadających im zmiennych decyzyjnych.

W przedstawionym algorytmie odcinek ma zawsze nachylenie nie większe niż jedność, co umożliwia skorzystanie z mechanizmu obliczeniowego algorytmu Bresenhama, charakteryzującego się wykorzystaniem reprezentacji liczb wyłącznie typu całkowitego oraz brakiem operacji dzielenia. W jednym z najlepszych znanych algorytmów [5] działających przy dowolnym nachyleniu odcinka, udało się co prawda



Fot. 1. Powiększony odcinek 0,4, czerwony – 15,11, zielony narysowany algorytmem *LineBC*

uniknąć reprezentacji liczb rzeczywistych, pozostała jednak operacja dzielenia całkowitego.

Przedstawiony algorytm LineBC ma niestety pewną niedogodność. Jeśli zmienną niezależną jest zmienna opisu koloru, a odcinek ma niewielką długość, to algorytm wyznacza kilkakrotnie te same współrzędne ekranowe x_e, y_e , zmieniając kolor piksela C . Może to spowodować zmianę koloru punktu początkowego C i narzuca konieczność przywrócenia mu zadanej wartości początkowej C_p po zakończeniu działania algorytmu. Tę sytuację rzadko spotyka się w praktyce. Typowym przypadkiem jest niewielka różnica koloru punktu początkowego i końcowego co powoduje, że zmienną niezależną jest jedna ze współrzędnych ekranowych.

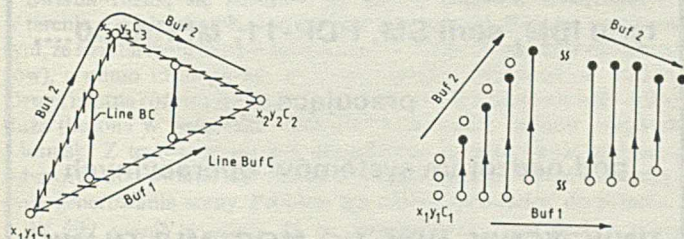
W przypadku stosowania tabeli kolorów należy umieścić w niej w obszarze $n_1 - n_2$ opis $(n_2 - n_1 + 1)$ kolorów będących wynikiem interpolacji między dwoma ustalonymi kolorami C_A i C_B . Sensowny wizualnie wynik działania algorytmu LineBC otrzymuje się gdy $n_p, n_k \in \langle n_1, n_2 \rangle$. Ilustrację działania algorytmu LineBC w postaci odcinka $(0,4) - (15,11)$ o czerwonym kolorze punktu początkowego i zielonym punktu końcowego przedstawiono na fot. 1. Zmiana koloru kolejnych pikseli jest ledwie dostrzegalna jako zmiana stopnia szarości na zamieszczonej czarno-białej fotografii.

Algorytm cieniowania wnętrza trójkąta

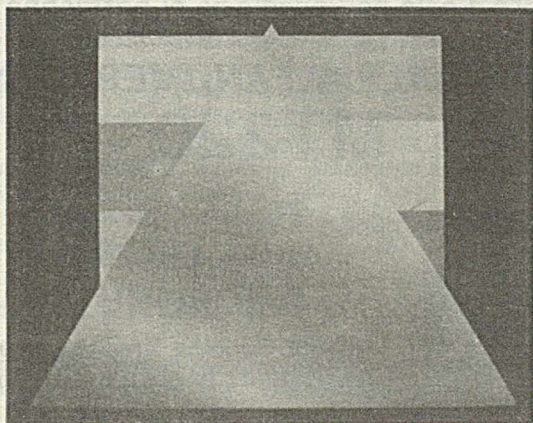
Algorytm cieniowania wnętrza trójkąta wywodzi się z klasy algorytmów *scan-line*. Jego istota polega na wypełnianiu wnętrza odcinkami wyłącznie poziomymi albo wyłącznie pionowymi. Kierunek wypełniania zależy od kształtu i orientacji trójkąta na ekranie.

W algorytmie *ShadeTriangle* wykorzystano rozbudowaną wersję algorytmu *LineBC* (nazwaną *LineBufC*), rysującego odcinek $(x_p, y_p, C_p) - (x_k, y_k, C_k)$. Algorytm *LineBufC* określa wartości współrzędnych ekranowych kolejnych pikseli odcinka oraz ich kolor $- x_i, y_i, C_i$ - i wpisuje piątkę (albo trójkę) liczb do bufora wyjściowego *Buf*. Jeśli zmienną niezależną jest zmienna opisu koloru, to wyliczone współrzędne pikseli (x_i, y_i, C_i) są wpisywane do pomocniczego bufora *AuxBuf*. Po wyznaczeniu współrzędnych wszystkich pikseli odcinka przegląda się bufor *AuxBuf* w celu wybrania kolejnych wartości wskazanej współrzędnej ekranowej x_e albo y_e i przepisania ich wraz z opisaniem koloru do bufora wyjściowego *Buf*. Operacja ta jest niezbędna do uzyskania w buforze wyjściowym *Buf* takiego ciągu współrzędnych ekranowych, w którym występują wszystkie kolejne wartości wskazanej współrzędnej.

Kierunek rysowania równoległych odcinków, którymi jest wypełnianie wnętrza trójkąta, zależy od orientacji najdłuższego boku. Długość



Rys. 1. Zasada cieniowania wnętrza trójkąta



Fot. 2. Trójkąt cieniowany za pomocą algorytmu *ShadeTriangle*; w tle paleta wykorzystanych barw

boku liczona w pikselach wynosi $L = \max(|x_k - x_p|, |y_k - y_p|) + 1$. Spośród współrzędnych ekranowych wyróżnia się tę, która ma większą różnicę współrzędnych punktu końcowego i początkowego:

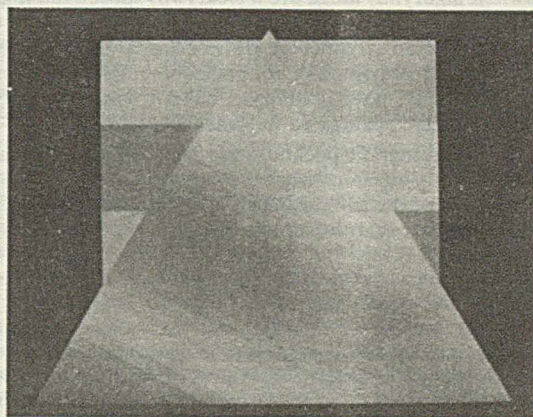
$$d = \begin{cases} x_e & \text{gdy } |x_k - x_p| \geq |y_k - y_p| \\ y_e & \text{w przeciwnym razie} \end{cases}$$

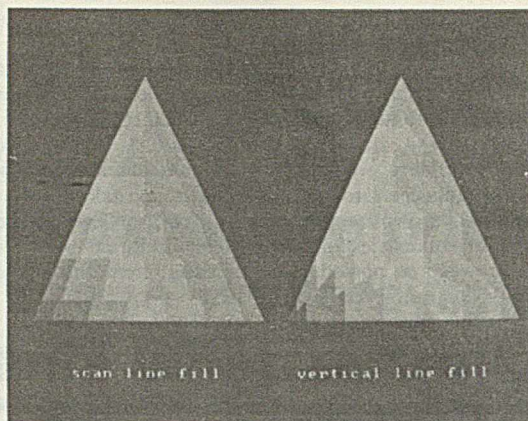
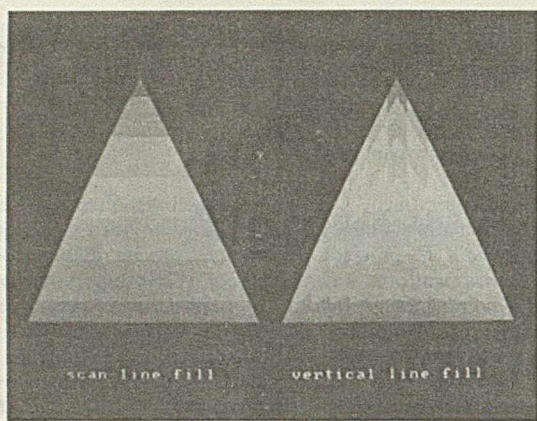
Wyznaczone algorytmem *LineBufC* wartości współrzędnych ekranowych i koloru dla pikseli leżących na najdłuższym boku są umieszczane w buforze *Buf1*. Współrzędne pikseli leżących na dwu pozostałych bokach są wprowadzane kolejno do *Buf2*. Jeśli algorytm *LineBufC* korzysta z pomocniczego bufora, to *AuxBuf* będzie przeglądany po współrzędnej d zarówno dla boku najdłuższego, jak i dla dwu pozostałych. Następnie są pobierane kolejne wartości współrzędnych i koloru z bufora *Buf1*. Bufor *Buf2* jest przeglądany do znalezienia pierwszego punktu o współrzędnej d równej współrzędnej pobranej z *Buf1*. Między punktami pobranymi z buforów *Buf1* i *Buf2* prowadzi się odcinek zgodnie z algorytmem *LineBC*. Szkic przedstawiający zasadę działania algorytmu *ShadeTriangle* przedstawiono na rys. 1, a wynik działania algorytmu cieniowania wnętrza przedstawiono na fot. 2. W tle jest widoczna paleta wykorzystywanych barw.

Kroki algorytmu *ShadeTriangle* $(x_1, y_1, C_1, x_2, y_2, C_2, x_3, y_3, C_3)$ są następujące:

- 1) wyznaczenie najdłuższego boku $L = \max(|x_k - x_p|, |y_k - y_p|) + 1$; przy równych różnicach decyduje kolejność boków 1-2, 2-3, 3-1;
- 2) dla najdłuższego boku wyznaczenie współrzędnej ekranowej d ; $d = x_e$, gdy $|x_k - x_p| \geq |y_k - y_p|$, $d = y_e$, gdy $|x_k - x_p| < |y_k - y_p|$;
- 3) wyznaczenie punktu początkowego na najdłuższym boku - wierzchołek o mniejszym numerze;
- 4) wprowadzenie zgodnie z algorytmem *LineBufC* współrzędnych kolejnych punktów najdłuższego boku, poczynając od wierzchołka wyznaczonego w kroku 3. do *Buf1*;
- 5) wprowadzenie zgodnie z algorytmem *LineBufC* współrzędnych punktów boku przyległego do najdłuższego, poczynając od wierzchołka wyznaczonego w kroku 3. do *Buf2*;
- 6) kontynuowanie wprowadzania (zgodnie z algorytmem *LineBufC*) współrzędnych punktów pozostałego boku, poczynając od wierzchołka będącego końcowym w kroku 5; (w krokach 4, 5, 6, ewentualne przeglądanie *AuxBuf* jest prowadzone po współrzędnej d wyznaczonej w kroku 2);
- 7) pobranie współrzędnych pikseli z *Buf1*;
- 8) przeszukanie *Buf1* aż do uzyskania zgodności współrzędnej d ze współrzędną pobraną z *Buf1*;
- 9) wykreślenie odcinka zgodnie z algorytmem *LineBC* między punktami o współrzędnych pobranych z *Buf1* i *Buf2*;
- 10) powtarzanie 7 - 9 aż do wyczerpania *Buf1*.

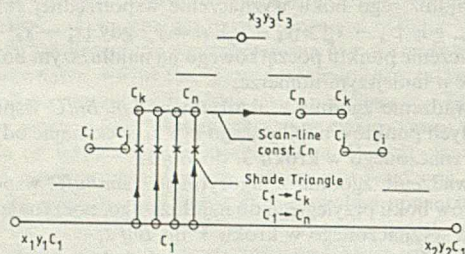
Zaletą przedstawionego algorytmu jest cieniowanie wnętrza odcinkami, których kierunek zależy od kształtu i orientacji trójkąta. Pozwala to uzyskać dokładniejsze rozmieszczenie koloru we wnętrzu trójkąta. Rozpatrzmy trójkąt $(x_1, y_1, C_1, x_2, y_2, C_2, x_3, y_3, C_3)$ przedstawiony na rys. 2. Rozkład koloru na bokach $x_1, y_1 - x_1, y_3$ oraz $x_2, y_2 - x_2, y_3$ jest identyczny. Klasyczny algorytm *scan-line* połączy punkty leżące na tych bokach poziomymi odcinkami o stałym kolorze. Algorytm *ShadeTriangle* wypełni taki trójkąt odcinkami pionowymi, dokonując interpolacji między





Fot. 3. Porównanie działania algorytmu ShadeTriangle – trójkąt wypełniony pionowo – z klasycznym algorytmem *scan-line*

kolorem C_1 a kolorem pikseli leżących na bokach $x_1y_1 - x_3y_3$ oraz $x_2y_2 - x_3y_3$. Kolor pikseli wyróżnionych krzyżykami, mających dla algorytmu *scan-line* stały kolor jest wynikiem interpolacji między kolorem C_1 a kolorami C_n do C_k . Porównanie efektów działania algorytmu *ShadeTriangle* i algorytmu *scan-line* przy identycznych danych wejściowych przedstawiono na fot. 3. Pełniejsza ilustracja wyników działania przedstawionych algorytmów jest zawarta w [4].



Rys. 2. Różnice w działaniu algorytmów cieniowania wnętrza trójkąta

★ ★ ★

W algorytmie kreślenia odcinka przyjęto zasadę wyznaczania współrzędnych punktów odcinka w przestrzeni pięcio- albo trójwymiarowej, w przeciwieństwie do znanych rozwiązań, w których odcinek jest rysowany na płaszczyźnie x_2y_2 i dla określonego piksla jest wyznaczany jego kolor. Znane podejście prowadzi do pojawiania się nachyleń odcinka większych niż 1, co typowo wymaga reprezentacji liczb rzeczywistych, a co najmniej wykonania operacji dzielenia [5]. W proponowanym podejściu nachylenie odcinka jest zawsze nie większe niż 1, co umożliwiło wykorzystanie mechanizmu obliczeniowego algorytmu *Bresenhama* do wyznaczania współrzędnych pikseli.

Istotną cechą algorytmu cieniowania wnętrza trójkąta jest uzależnienie kierunku wypełniania od kształtu i orientacji trójkąta na ekranie. Przedstawione algorytmy mogą być w efektywny sposób zaimplementowane w strukturach sprzętowych, ze względu na wykorzystywanie najprostszych działań arytmetyki dwójkowej. Działanie algorytmu można by wówczas przyspieszyć dokonując współbieżnego wyliczania współrzędnych pikseli na najdłuższym boku i dwu pozostałych, co pozwoliłoby uniknąć operacji na buforach *Buf1* i *Buf2*.

Przedstawione fotografie pochodzą ze zdjęć ekranu monitora o wielkości 384×288 j.v. wyposażonego w tabelę kolorów o 256 pozycjach i 6-bitowych konwerterach wizyjnych. Ze względów poligraficznych zamieszczono zdjęcia czarno-białe.

LITERATURA

- [1] Bresenham J.E.: Run Length Slice Algorithm for Incremental Lines. In: Earnshaw R.A.: Fundamental Algorithms for Computer Graphics, pp. 59–104, Springer Verlag, Berlin 1985
- [2] Gouraud H.: Continuous shading of curved surfaces. IEEE Transactions on Computers, Vol. C-20, No. 6, pp. 623–629, June 1971
- [3] Rzeszut J.: Efekt perspektywy powietrznej w generowaniu obrazu dla symulatorów. INFORMATYKA 1991

- [4] Rzeszut J.: Operacje w przestrzeniach kolorów dla potrzeb cyfrowej generacji obrazu. Rozprawa doktorska. Politechnika Warszawska 1989
- [5] Swanson R.W., Thayer L.J.: A Fast Shaded-Polygon Renderer. Computer Graphics. Vol. 20, No. 4, pp. 95–101, 1986.

Wszystkich zainteresowanych nabyciem INFORMATYKI prosimy o kontakt z redakcją (tel. 39-14-34) lub Działem Handlowym SIGMY (tel. 40-30-86) – możemy wysłać egzemplarze za zaliczeniem pocztowym. Czytelnicy z terenu Warszawy mogą kupić poszczególne numery w Klubie SIGMY przy ul. Mazowieckiej 12, w księgarni PP „Domu Książki” przy ul. Mokotowskiej 51/53 oraz w lokalu redakcji.

Jednocześnie przypominamy, że najpewniejszą formą otrzymywania INFORMATYKI jest prenumerata.

Posiadasz maszyny

typu IBM, serii SM, PDP-11, MERA400...,

pracujące

pod nadzorem systemów operacyjnych

UNIX, XENIX, RSX, PC-MOS, MULTILINK,
CONCURRENT PCDOS, QNX, CROOK...

To z całą pewnością są ci potrzebne

TERMINALE ALFANUMERYCZNE
MT220

Skontaktuj się z nami.



MICRONET

0/17/90

Zakłady Elektroniczne

"MICRONET"

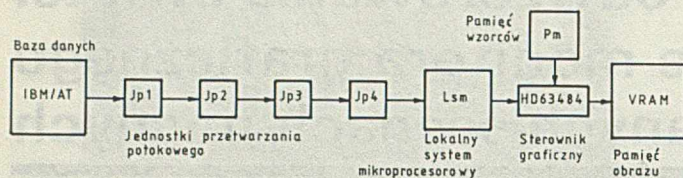
81-836 Sopot ul. Krasickiego 9
tel. 51-13-17 tlx 051-2876 mcrnt

ZAPRASZAMY

17

Realizacja techniczna

W celu wyświetlania rzutów światła na ekranie monitora opracowano specjalny tor (rys.). Składa się on z czterech identycznych jednostek przetwarzających dane w sposób potokowy, lokalnego 16-bitowego systemu mikroprocesorowego, sterownika graficznego HD63484 [2] oraz pamięci wzorców. W każdej jednostce przetwarzania potokowego wykorzystano mikroprocesor sygnałowy.



Schemat blokowy toru generacji rzutów światła

Informacje o położeniu obserwatora oraz opisy światła tor otrzymuje z bazy danych, przechowywanej w pamięci komputera IBM/PC/AT. Opisy światła obejmują: położenie światła w terenie, kolor, charakterystyki kierunkowe i sposób migotania [4]. Co 33 ms komputer umieszcza aktualne dane w dwudostępnej pamięci buforowej, skąd pobiera je pierwsza jednostka przetwarzająca. Jednostki przetwarzania potokowego dokonują transformacji macierzowych, obcinania, wyznaczania średnicy i współrzędnych rzutów światła na ekranie. Lokalny system mikroprocesorowy odbiera strumień opisów rzutów światła na ekranie i na ich podstawie programuje sterownik graficzny. Z uwagi na fakt, że wielkość rzutu światła może być mniejsza niż jeden piksel, lokalny system mikroprocesorowy dokonuje translacji takich rzutów na rzuty jednopikselowe zgodnie z zasadą, że jasność światła jest odwrotnie proporcjonalna do kwadratu odległości od obserwatora. Sterownik graficzny kopiuje wzorce rzutów światła do odpowiednich obszarów pamięci obrazu. Pamięć obrazu jest zdublowana i pracuje w układzie „ping-pong”, tzn. w czasie zapełniania jednego bloku na ekran jest wyprowadzana zawartość drugiego. Zamiana bloków miejscami następuje co 33 ms. Zastosowanie układów pamięci typu VideoRAM uwalnia sterownik graficzny od ciągłego generowania adresów odczytu i w ten sposób przyspiesza wyprowadzanie informacji.

LITERATURA

- [1] Hearn D., Baker M.: Grafika komputerowa. WNT, Warszawa 1990
- [2] HD63484 Advanced CRT Controller. Hitachi Data Sheet, 1985
- [3] Stępień C.: Graphics Controllers in Microprocessor Systems – raport wewnętrzny
- [4] Aneks nr 14 do konwencji dotyczącej międzynarodowego lotnictwa cywilnego. Międzynarodowa Organizacja Lotnictwa Cywilnego, wyd. 8, marzec 1983.

Efekt perspektywy powietrznej w generowaniu obrazu dla symulatorów

dokończenie ze s. 21

LITERATURA

- [1] Ablamowicz A.: Loty w trudnych warunkach atmosferycznych – IFR. WKiŁ, Warszawa, 1982
- [2] Babienko W. C.: Imitatory wizualnej obstanowki trenazherov letatielnych apparatov. Maszynostrojienie, Moskwa, 1978
- [3] Gavrilow V. A.: Proznacnost' atmosfery i widimost'. Gidrometeoizdat Leningrad, 1966
- [4] Lansdown J.: Visual perception and computer graphics. In: Earnshaw R. E.: Fundamental algorithms for computer graphics. pp. 1005–1026, Springer, 1985
- [5] Minnaert M.: Svet i cviet v prirode. Gos. Izd. Fiziko-matemat. lit. Moskwa, 1958
- [6] Nishita T., Nakamae E.: Continuous tone representation of three-dimensional objects illuminated by sky light. Computer Graphics. Vol. 20. No. 4, pp. 125–132, 1986
- [7] Rzeszut J.: Operacje w przestrzeniach kolorów dla potrzeb cyfrowej generacji obrazu. Rozprawa doktorska. Politechnika Warsz. 1989
- [8] Schachter B. J.: Computer image generation. John Wiley & Sons, New York, 1983
- [9] Tverskij P. N. et al.: Kurs meteorologii. Gidrometeoizdat, Leningrad, 1951
- [10] Wyszecki G., Stiles W. S.: Color Science-Concepts and methods. Quantitative data and formulae. 2nd Ed. John Wiley & Sons, New York, 1982.



SWEDEX UNIVERSAL

Ltd

oferuje:

MODEMY DUPLEKSOWE DO TRANSMISJI DANYCH:

- ☐ pracujące na łączach telefonicznych komutowanych i dzierżawionych, jednoparowych, zgodnie z normami V.21, V.22, V.22 bis CCITT:
MODEM 1230 P (0–300 b/s asynchron., 600/1200 b/s asynchron./synchron.)
MODEM 2412 P (0–300 b/s asynchron., 1200/2400 b/s asynchron./synchron.)
MODEM 2412 H (automatyczny – standard Hayes, 0–300 b/s asynchron., 1200/2400 b/s asynchron./synchron.);
- ☐ pracujące na łączach telefonicznych dzierżawionych, dwuparowych: Konwerter KN-9600 (modem typu baseband, 2400 4800 9600 b/s synchron.);
- ☐ pracujące na łączach telefonicznych dzierżawionych, jednoparowych.

Wszystkie modemy posiadają świadectwo homologacji Instytutu Łączności

- ☐ Konwerter RS-232 na pętłę prądową 20 mA (do 19 200 b/s asynchron./synchron., odległość do 2 km).

DO KOMPUTERÓW PC XT/AT:

- ☐ interfejsy pomiarowe IEC 625 (IEEE 488):
SU-GPIB (do 10 kB/s),
SU-GPIBF (do 500 kB/s);
- ☐ programator SU-PROG pamięci EPROM 2716-27512 i mikrokomputerów jednoukładowych MCS 48, MCS 51;
- ☐ pakiet transmisji synchronicznej BSC: umożliwia pracę IBM PC jako terminala ICL 7181, ICL 7020, IBM 2780, IBM 3780, IBM 3270;
- ☐ oprogramowanie emulacyjne stacji IBM 3780.

DO MERY 9150:

- ☐ Koncentrator Stanowisk Operatorskich KSO-4;
- ☐ Adapter Wyniesionego Stanowiska Operatorskiego ASO z modemami;
- ☐ Adapter Telegraficzny ATG-2.

DO EMC ODRA 1305:

- ☐ Adapter Telekomunikacyjny AT-8/MPX (zastępuje UPD)

TESTERY:

- ☐ KSO (systemy mikroprocesorowe na INTEL 8080 i 8085);
- ☐ V.24 (zgodny z CCITT V.24/V.28 i EIA RS-232 C).

OFERTY – ZAMÓWIENIA – INFORMACJE

SWEDEX UNIVERSAL Ltd
ul. Wspólna 1,
05-075 Wesoła k. Warszawy
teleks 813935 swedx pl,
Warszawa tel. (022)15-33-65

ZO/236/90



SWEDEX UNIVERSAL

Efekt perspektywy powietrznej w generowaniu obrazu dla symulatorów

Obraz generowany na ekranie monitora w symulatorze powinien dostarczać obserwatorowi wrażeń umożliwiających orientację w przestrzeni. W symulatorach lotu celowe jest uwzględnianie efektów wizualnych wynikających z warunków pogodowych oraz pory dnia [2, 8]. Jednym ze zjawisk potęgujących wrażenie przestrzenności obrazu jest perspektywa powietrzna [4]. Zjawisko to polega na zmianie barwy i jasności obiektów oddległych od obserwatora jest powodowane przez rozpraszanie światła przez atmosferę i zawarte w niej cząsteczki wody. W symulacji warunków pogodowych istotna jest symulacja zmian odległości widzialności powodowanych przez obecność mgły. Celowe jest ponadto wprowadzenie efektów wynikających ze zmian rozproszonego oświetlenia terenu, umożliwiających symulację zjawisk zmierzchowych i obrazów nocnych.

Problem perspektywy powietrznej nie jest programem nowym. Pojawił się w meteorologii i w nauce o widzialności w atmosferze [3]. Obie te dziedziny zajmowały się wyłącznie jasnością (a w zasadzie strumieniem światła) nie uwzględniając zagadnień barwy. Z kolei w grafice komputerowej spotyka się problem efektu mgły [6] wyłącznie w postaci równania (20). W artykule przedstawiono próbę całościowego potraktowania problemu, wychodzącą od faktów związanych z widzialnością w atmosferze, z przejściem na zjawiska barwne, zakończoną omówieniem procesu generowania obrazu [7].

Zjawisko perspektywy powietrznej

Perspektywa powietrzna jest zjawiskiem powodującym upodabnianie barw oddległych przedmiotów do barwy nieba widocznego nad linią horyzontu. Jasnoniebieska barwa czystego nieba jest efektem rozpraszania molekularnego na cząsteczkach powietrza proporcjonalnego do $1/\lambda^4$ [9], przy czym λ jest długością fali promieniowania świetlnego, co oznacza silniejsze rozpraszanie krótkofalowej części widma. Grubość rozpraszającej warstwy powietrza można oszacować na około 335 km, przyjmując za [5] zredukowaną wysokość atmosfery wynoszącą, 8,8 km. Wartość 335 km została wyznaczona jako promień widoczności z wysokości 8,8 km, przy uwzględnieniu krzywizny powierzchni Ziemi. Przy zwiększaniu liczby i rozmiarów cząsteczek wody w atmosferze rozpraszanie staje się bardziej równomierne. Barwa światła rozproszonego zaczyna zmieniać się od jasnoniebieskiej do białej, a oddległe obiekty zaczynają znikać w białej mgłę.

Zjawisko wtapiania się obiektu w tło jest związane z pojęciem kontrastu między obiektem a tłem oraz z kontrastem progowym, określającym minimalną postrzegalną różnicę jasności. Znanie źródła dotyczące widzialności w atmosferze [3] nie uwzględniają barwy obiektu, posługując się wyłącznie kontrastem jasności K odnoszącym się do różnicy strumieni światła Φ

$$K = (\Phi_2 - \Phi_1)/\Phi_2 \quad \text{dla } \Phi_2 \geq \Phi_1 \quad (1)$$

$$K = \begin{cases} 1 - \Phi_0/\Phi_B & \Phi_B \geq \Phi_0 \\ 1 - \Phi_B/\Phi_0 & \Phi_0 > \Phi_B \end{cases} \quad (2)$$

Przy czym: Φ_0 – strumień światła obiektu, Φ_B – strumień światła tła.

Wrażenie jasności jest związane z wielkością strumienia światła. W tym artykule używa się pojęcia jasności w sensie wielkości strumienia lub sumy strumieni światła.

Kontrast progowy ε przy którym obiekt wtapia się w tło, bardzo zależy od rozmiarów kątowych obiektu. Dla obiektów większych niż $20'$ stabilizuje się na poziomie 0,02 [3]. W celu określenia obserwowanej jasności obiektu przyjmuje się prosty model rozpraszania światła w warstwie mgły znajdującej się między obiektem a obserwatorem. Jasność cienkiej warstwy mgły oświetlonej od góry strumieniem światła Φ zależy od kąta padania promieni Θ oraz jest funkcją gęstości i wielkości cząsteczek rozpraszających $p(\Theta)$. Nie jest uwzględniane tłumienie oświetlenia w pionie. Rozwiązanie prostego równania różniczkowego daje w wyniku wzór na strumień światła Φ_f rozprazanego w kierunku obserwatora.

$$\Phi_f = \frac{\Phi p(\Theta)}{\alpha} (1 - e^{-\alpha L}) \quad (3)$$

przy czym: L – długość warstwy mgły, α – współczynnik tłumienia.

Wykładniczo tłumiony strumień światła odbitego od obiektu Φ_{OB} dociera do obserwatora jako Φ_O :

$$\Phi_O = \Phi_{OB} e^{-\alpha L_O} \quad (4)$$

gdzie L_O odległość obserwator-obiekt.

W ten sam sposób jest tłumiony strumień światła nieba. Zakłada się, że strumień światła nieba Φ_{SH} pochodzi z rozpraszania światła słonecznego w atmosferze na odległości od linii horyzontu do granicy atmosfery i jego wykładniczo tłumiony w warstwie mgły między linią horyzontu a obserwatorem, docierając do niego jako Φ_s

$$\Phi_s = \Phi_{SH} e^{-\alpha L_h} \quad (5)$$

przy czym L_h – odległość linii horyzontu od obserwatora.

Na obserwowaną jasność obiektu Φ_{OP} składa się tłumiony strumień światła pochodzącego od obiektu Φ_O oraz światła Φ_{fO} pochodzące z warstwy mgły między obiektem a obserwatorem:

$$\Phi_{OP} = \Phi_O + \Phi_{fO} \quad (6)$$

$$\Phi_{OP} = \Phi_{OB} e^{-\alpha L_O} + \frac{\Phi p(\Theta)}{\alpha} (1 - e^{-\alpha L_O}) \quad (7)$$

Obserwowana jasność nieba nad linią horyzontu Φ_{SP} jest sumą tłumionego strumienia światła nieba Φ_s oraz światła Φ_{fH} pochodzącego z rozpraszania we mgłę między obserwatorem a linią horyzontu:

$$\Phi_{SP} = \Phi_s + \Phi_{fH} \quad (8)$$

$$\Phi_{SP} = \Phi_{SH} e^{-\alpha L_h} + \frac{\Phi p(\Theta)}{\alpha} (1 - e^{-\alpha L_h}) \quad (9)$$

Przy dużym współczynniku tłumienia α za jasność nieba nad horyzontem można przyjąć jasność warstwy mgły między obserwatorem a linią horyzontu Φ_{SPH} :

$$\Phi_{SPH} \approx \Phi_{fH} = \frac{\Phi p(\Theta)}{\alpha} (1 - e^{-\alpha L_h}) \approx \frac{\Phi p(\Theta)}{\alpha} \quad (10)$$

W takich warunkach wzór na obserwowaną jasność obiektu Φ_{OP} przyjmuje postać

$$\Phi_{OP} = \Phi_{OB} e^{-\alpha L_O} + \Phi_{SPH} (1 - e^{-\alpha L_O}) \quad (11)$$

Można założyć, że przy dużym współczynniku tłumienia α rozpraszanie jest równomierne i zależności (9) i (11) będą słuszne również dla koloru $C[r', g', b']$, traktowanego jako wektor składowych trójchromatycznych r', g', b' .

$$C_{SP} = C_{SH} e^{-\alpha L_H} + C_{SPH} (1 - e^{-\alpha L_H}) \quad (12)$$

$$C_{OP} = C_{OB} e^{-\alpha L_O} + C_{SPH} (1 - e^{-\alpha L_O}) \quad (13)$$

Dokładne obliczenie wartości składowych r', g', b' wymagałoby znajomości widmowego rozkładu energii światła słonecznego $\Phi(\lambda)$ i zależności współczynnika tłumienia od długości fali $\alpha(\lambda)$ oraz przeprowadzenia pełnej procedury wyznaczania wartości składowych trójchromatycznych [10].

Meteorologiczna odległość widzialności

Dysponując wzorem (11) można określić odległość widzialności L_r danego obiektu O na tle B o większej jaskrawości.

$$K_o = 1 - \Phi_{OB}/\Phi_{BB} \quad (14)$$

przy czym Φ_{OB} – strumień światła odbity od obiektu, Φ_{BB} – strumień światła odbity od tła,

$$L_r = \frac{1}{\alpha} \ln \left[\frac{K_o}{\varepsilon} \frac{\Phi_{BB}}{\Phi_{SPH}} + \left(1 - \frac{\Phi_{BB}}{\Phi_{SPH}} \right) \right] \quad (15)$$

przy czym ε – kontrast progowy.

Z wzoru (15) można określić tzw. meteorologiczną odległość widzialności L_M . Jest to odległość, z której widać czarny obiekt o rozmiarach kątowych większych niż $20'$ na tle nieba nad horyzontem [3]. Przyjmując, że dla czarnego obiektu $\Phi_{OB} = 0$, otrzymamy kontrast K_o wartości równej 1, a jaskrawość tła Φ_{BB} jest równa jaskrawości nieba Φ_{SPH} . Stąd

$$L_M = \frac{1}{\alpha} \ln \frac{1}{\varepsilon} \quad (16)$$

Wartość kontrastu progowego ε dla obiektu o rozmiarach kątowych przekraczających $20'$ przyjmuje się jako 0,02. Stąd:

$$L_M = \frac{1}{\alpha} \cdot 3,91 \quad (17)$$

Meteorologiczna odległość widzialności jest stosowanym na świecie pojęciem określającym warunki widzialności na lotniskach [1, 3].

Metoda implementacji efektu perspektywy powietrzeń

Wychodząc z podanych wyżej zależności należy ustalić reguły określania barwy i jaskrawości nieba oraz widocznych obiektów terenu. Celowe jest jednolite ujęcie zjawisk charakterystycznych zarówno dla małych, jak i dla dużych wartości współczynnika tłumienia α . Fizyczna interpretacja zjawiska prowadzi do wniosku, że w zakresie małych wartości α istnieje zależność od długości fali $\alpha(\lambda)$ zanikająca przy zwiększaniu wartości α . Dla dużych wartości $\alpha(\lambda) = \text{const}$, co oznacza równomierne rozpraszanie światła, a więc białą barwę światła rozpraszanego. Potwierdzenie w postaci orientacyjnych wykresów ilustrujących charakter zjawiska można znaleźć w [9], brak jest natomiast danych liczbowych. Zaproponowane poniżej podejście jest próbą uwzględnienia efektów zarówno nieznacznego rozmycia barw obiektów kierunku barwy jasnoniebieskiej, jak i wtapienia się obiektów w białą mgłę, przy braku zależności współczynnika tłumienia α od długości fali λ .

Zadawanie warunków pogodowych wymaga podania wartości meteorologicznej odległości widzialności L_M . Wartość współczynnika tłumienia α wyznacza się w zależności:

$$\alpha = 3,91/L_M \quad (18)$$

Obserwowaną barwę i jaskrawość nieba nad linią horyzontu wyznacza się według wzoru (12):

$$(C_{SP} = C_{SH} e^{-\alpha L_H} + C_{SPH} (1 - e^{-\alpha L_H})) \quad (19)$$

C_{SH} jest kolorem czystego nieba o współrzędnych trójchromatycznych $x = 0,247$ $y = 0,251$; C_{SPH} jest kolorem w pełni zachmurzonego nieba o barwie mającej współrzędne trójchromatyczne $x = 0,313$ $y = 0,329$, a więc barwie białej (jest to barwa biała D_{6500} [10]).

Celowe jest wprowadzenie zmian koloru nieba wraz ze zwiększaniem wysokości nad horyzontem. Ścisły związek określający zmianę koloru od C_{SP} do C_{SH} wymaga uwzględnienia zmian gęstości mgły wraz z wysokością.

Przyjęcie dla określenia obserwowanego koloru obiektu C_{OP} wzoru (13) wyprowadzonego dla dużych wartości współczynnika α prowadziłoby każdorazowo do rozmywania obiektu w kierunku białego koloru C_{SPH} , bez możliwości uzyskania efektów charakterystycznych dla małych wartości α . Wobec powyższego proponuje się, by rozmywanie koloru obiektu zachodziło nie w kierunku koloru C_{SPH} , ale w kierunku obliczonego według wzoru (19) koloru nieba nad horyzontem C_{SP} .

$$C_{OP} = C_{OB} e^{-\alpha L_O} + C_{SP} (1 - e^{-\alpha L_O}) \quad (20)$$

Wzór (20) zachowuje fizyczną istotę zjawiska. Wprowadzenie rozmycia w kierunku koloru C_{SP} w pewnym stopniu odzwierciedla zależność $\alpha(\lambda)$ dla małych wartości α .

Kolor C_{OB} jest wyznaczany z uwzględnieniem barwy i natężenia oświetlenia terenu. Informacja o barwie obiektu jest przechowywana w bazie danych w postaci trójki liczb reprezentujących względne współczynniki odbicia $\rho(R)$, $\rho(G)$, $\rho(B)$ dla światła o barwach R , G i B . Rozproszone oświetlenie terenu S_d jest opisane przez podanie wartości trzech składowych:

$$S_d = \begin{bmatrix} S_d(R) \\ S_d(G) \\ S_d(B) \end{bmatrix} \quad S_d(k) < 0,1 > \quad (21)$$

Kolor obiektu C_{OBd} wynikający z oświetlenia światłem rozproszonym, wyrażony w wartościach składowych r', g', b' ma postać:

$$C_{OBd} = \begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} S_d(R) \cdot \rho(R) \\ S_d(G) \cdot \rho(G) \\ S_d(B) \cdot \rho(B) \end{bmatrix} \quad (22)$$

Przy założeniu, że obiekt odbija światło w sposób rozproszony, można uwzględnić kierunkowe oświetlenie terenu S_s . Wymaga to znajomości kąta padania Θ promieni światła na daną powierzchnię oraz kąta ψ między normalną do powierzchni a prostą łączącą oko obserwatora z obiektem. Kolor obiektu C_{OB} , wynikający z oświetlenia światłem kierunkowym ma postać:

$$C_{OB} = \begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} (S_s(R) \cdot \cos \Theta) \cdot \rho(R) \cdot \cos \psi \\ (S_s(G) \cdot \cos \Theta) \cdot \rho(G) \cdot \cos \psi \\ (S_s(B) \cdot \cos \Theta) \cdot \rho(B) \cdot \cos \psi \end{bmatrix} \quad (23)$$

Wartość kąta Θ spoza zakresu $\pm 90^\circ$ oznacza, że dana powierzchnia nie jest oświetlana przez źródło kierunkowe S_s . Wartość kąta ψ spoza zakresu $\pm 90^\circ$ oznacza, że dana powierzchnia nie jest widoczna dla obserwatora. Światło odbite od obiektu jest sumą światła odbicia oświetlenia rozproszonego C_{OBd} oraz odbicia oświetlenia kierunkowego C_{OBs} .

$$C_{OB} = C_{OBd} + C_{OBs} \quad (24)$$

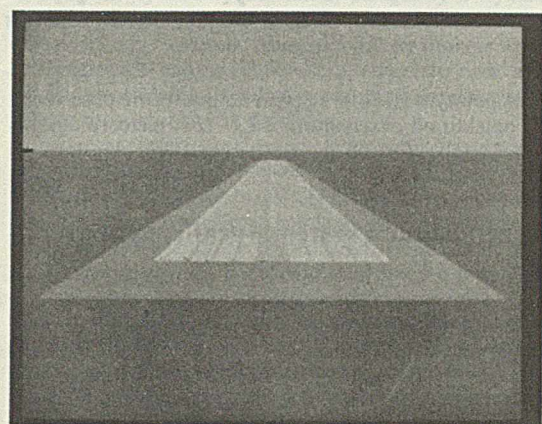
Przedstawiony powyżej sposób określania koloru obiektu umożliwia uwzględnienie w generowanym obrazie efektów wynikających ze zmian natężenia i barwy oświetlenia krajobrazu, np. symulację scen zmierzcho-wych. Pełne uwzględnienie naturalnego oświetlenia wymaga znacznie większego nakładu obliczeń [6].

Operacje na barwach obiektów stanowią jeden z końcowych etapów generowania obrazu. Obraz jest tworzony na podstawie opisu terenu w postaci zbioru wielokątów o określonych współrzędnych wierzchołków oraz współczynników odbicia ρ reprezentujących barwę wielokąta, zadanych warunków oświetlenia, meteorologicznej odległości widzialności oraz zestawu parametrów opisujących położenie obserwatora w przestrzeni. Nie wdając się w szczegóły dotyczące transformacji geometrycznych można przyjąć, że ich wynikiem są współrzędne ekranowe wierzchołków wielokąta i ich odległości od obserwatora. Ze względu na niewielki kąt widzenia wynikający z rozmiarów ekranu

rzeczywistą odległość L_i wierzchołka i przybliża się wartością współrzędnej Z_i

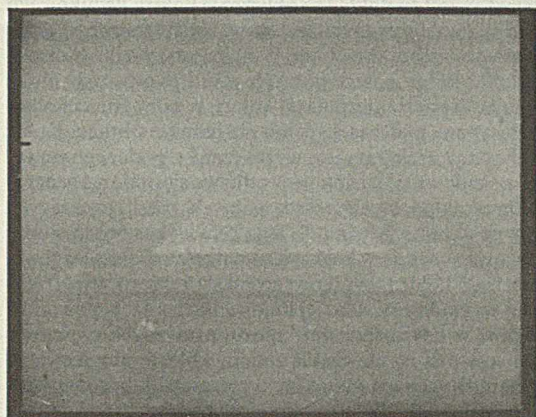
$$Z_i \approx L_i = \sqrt{X_i^2 + Y_i^2 + Z_i^2} \quad (25)$$

przy czym X_i , Y_i , Z_i są współrzędnymi wierzchołka i w układzie współrzędnych obserwatora. Zadane warunki oświetlenia pozwalają wyznaczyć kolor obiektu C_{OB} zgodnie ze wzorami (21), (22), (23). Kolor wierzchołków wielokąta wyznacza się według wzorów (19) i (20). Następnie należy wypełnić wnętrze wielokąta dokonując interpolacji między kolorami wierzchołków za pomocą algorytmu cieniowania wnętrza wielokąta. Na fotografiach przedstawiono widok pasa startowego z odległości 300 m przed progiem pasa i wysokości 17 m, przy różnych wartościach meteorologicznej odległości widzialności.



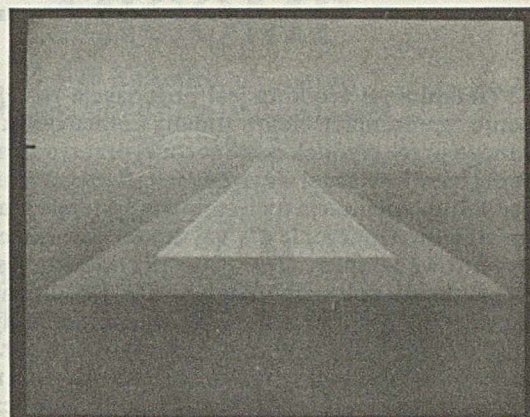
a)

$L_m = 200 \text{ km}$



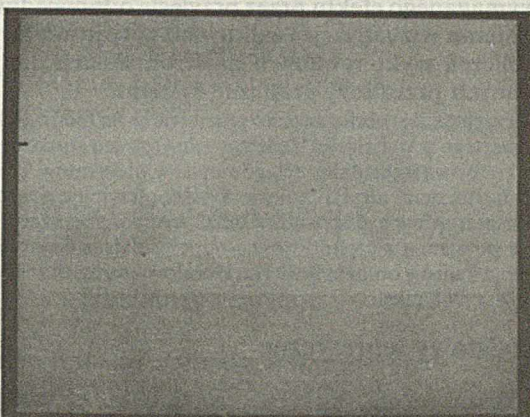
c)

$L_m = 1 \text{ km}$



b)

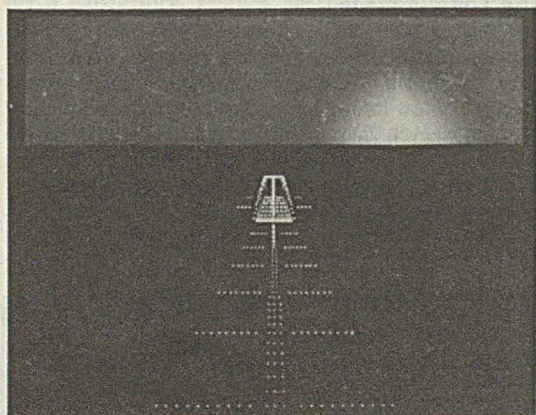
$L_m = 5 \text{ km}$



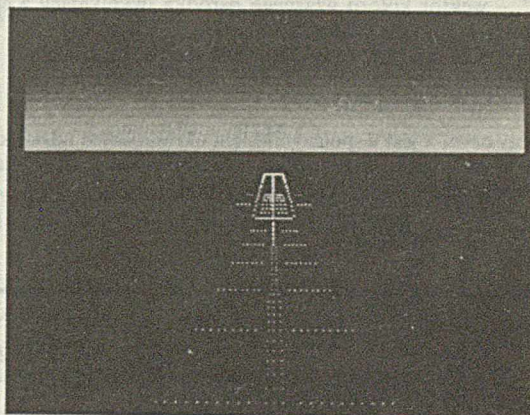
d)

$L_m = 600 \text{ m}$

Fot. 1. Widok pasa startowego przy zmniejszającej się meteorologicznej odległości widzialności



a)



b)

Fot. 2. Sceny zmierzchowe

dokończenie na s. 18

Realizacja efektu zmiany koloru obiektu w ośrodku optycznym w systemie generowania obrazu w czasie rzeczywistym

Przedmiotem artykułu jest propozycja rozwiązania problemu reprezentacji efektu zmiany koloru obiektu obserwowanego przez ośrodek o skończonej przezroczystości i określonej barwie własnej w systemie graficznym czasu rzeczywistego. Implementacja tego efektu w systemie graficznym ma podstawowe znaczenie dla uzyskania naturalnego wyglądu prezentowanej sceny. Typowym przykładem jest tu generowanie obrazu powierzchni terenu z uwzględnieniem wpływu przezroczystości powietrza, np. w postaci mgły. Warunek generowania obrazu w czasie rzeczywistym (tj. 30 kadrów na sekundę) wyklucza możliwość zastosowania rozwiązania analitycznego i wykonywania dokładnych obliczeń wynikowego koloru każdego piksla obrazu w trakcie jego generowania. Proponowane rozwiązanie umożliwia uzyskanie wymaganego efektu przez przyjęcie odpowiedniej struktury systemu wizualizacji i całkowicie sprzętową implementację obliczeń wykonywanych na strumieniach danych generowanych przez tory graficzne systemu.

Przyjęto, że obraz będzie generowany na podstawie opisu sceny zawartego w wejściowej bazie danych. Opis ten strukturę hierarchiczną (warstwową) i składa się z ciągów opisów wielokątów. Z wielokątem jest związany m.in. atrybut koloru. System otrzymuje ponadto informacje o współrzędnych obserwatora oraz kierunku obserwacji. Informacja ta jest pobierana z częstotliwością odpowiadającą częstotliwości generowania kadrów obrazu, tj. 30 Hz. Docelowa rozdzielczość obrazu wynosi 1024×768 piksli.

Podstawy teoretyczne

Zmiana koloru obserwowanego obiektu może być przedstawiona jako mieszanie addytywne barw. Przyjmując oznaczenia:

C – wektor składowych trójkromatycznych koloru wynikowego $[r', g', b']$,

C_o – wektor koloru wyjściowego,

C_e – wektor koloru własnego ośrodka, wynikający z charakterystyki widmowej rozpraszania ośrodka, można tę operację opisać wzorem interpolacyjnym:

$$C = C_o \star x + C_e \star (1 - x)$$

przy czym x jest współczynnikiem określającym udział koloru wyjściowego w kolorze wynikowym. Dla omawianego zjawiska wartość tego współczynnika jest określana wzorem

$$x = \exp(-k \star l/l_m)$$

przy czym k jest współczynnikiem przyjętym na podstawie obserwacji, l jest odległością obiektu od obserwatora a l_m – tzw. meteorologiczną odległością widzenia [1]. Wartość l_m jest parametrem określającym przezroczystość ośrodka.

Rozwiązanie problemu w systemie graficznym o typowej strukturze

Przy założeniu, że system graficzny jest zakończony buforem pamięci obrazu zawierającym wartości trzech barw składowych dla każdego piksla, realizacja efektu zmiany koloru wymaga obliczenia koloru każdego piksla reprezentującego dany wielokąt. Jak już wspomniano, realizacja analityczna, ze względu na wymóg pracy w czasie rzeczywistym, nie jest możliwa. Stosuje się więc mechanizmy umożliwiające uzyskanie wynikowego koloru piksla w sposób przybliżony. Jednym z możliwych rozwiązań jest tu dokładne obliczenie koloru wynikowego wierzchołków wielokąta, a następnie zastosowanie sprzętowego gładkiego cieniowania powierzchni wielokąta. W najprostszym do realizacji sprzętowej przypadku kolor poszczególnych piksli tworzących obraz wielokąta otrzymuje się przez interpolację koloru między wierzchołkami wielokąta, realizowaną podczas wypełniania pamięci obrazu. Kolor wynikowy wierzchołków wielokąta jest wyznaczany z podanego powyżej wzoru. Liniowe cieniowanie wielokątów odbywa się podczas generowania wartości barw składowych poszczególnych piksli tworzących obraz wielokątów na ekranie. Można do tego celu wykorzystać moduł mikroprogramowany zawierający kaskadę sumatorów. Ponieważ funkcja opisująca zależność koloru wynikowego piksla od jego współrzędnych ma charakter wykładniczy, więc jej liniowa interpolacja powoduje powstawanie błędów w odwzorowaniu koloru, usuwanych częściowo przez małą wrażliwość oka na niewielkie zmiany koloru oraz niewielki udział w obrazowanych scenach obiektów o rozmiarach porównywalnych z l_m . Użycie takich mechanizmów ogranicza również zastosowania systemu do obrazowania scen w warunkach stosunkowo dobrej widoczności.

Proponowane rozwiązanie

Proponowane rozwiązanie, będące rozwinięciem koncepcji zaproponowanej przez dra inż. J. Rzeszuta, zakłada wykorzystanie do generowania koloru wynikowego obiektów informacji pochodzącej z dwóch planów pamięci ekranu, stanowiących zakończenia dwóch torów graficznych systemu. Wartości barw składowych piksla są wyznaczane w czasie wyprowadzania danych z planów pamięci obrazu do wyświetlania. Jeden z planów pamięci obrazu, zwany platem atrybutów, zawiera m.in. informację o kolorze wyjściowym poszczególnych piksli. W drugim

Uwaga!

Wszystkie sprawy ogłoszeniowe są załatwiane bezpośrednio w Redakcji.
Mimo podwyższenia ceny ogłoszeń są one jeszcze tanie!

z wykorzystywanych planów – planie odległości – przechowuje się informację o odległości obiektów sceny reprezentowanych przez poszczególne piksele ekranu od obserwatora; stanowi on modyfikację klasycznego bufora odległości (ang. *z-buffer*). Kolor wynikowy piksela uzyskuje się przez realizację operacji arytmetycznych wynikających z przedstawionego wcześniej wzoru.

Postać informacji przechowywanej w planie odległości musi zapewniać reprezentację przedziału odległości w granicach od l_{min} do nieskończoności, przy czym wymagana precyzja reprezentacji jest odwrotnie proporcjonalna do odległości. Dotychczasowe prace wskazują na decydujące znaczenie postaci informacji przechowywanej w planie odległości dla wierności odwzorowania koloru wynikowego. Z punktu widzenia implementacji systemu wizualizacji korzystne jest reprezentowanie wartości odległości jak najkrótszym słowem danych. Stwierdzono, że przechowywana wielkość nie może być wartością odległości ze względu na jej liniowy charakter, niezgodny z wykładniczą postacią funkcji odwzorowania. Rozważa się możliwość wypełniania planu odległości odwrotnością odległości lub jej funkcję wykładniczą.

Wydzielenie planu atrybutów pozwala zastosować w torze atrybutów układy wypełniania wielokątów stałym kolorem, charakteryzujące się bardzo dużą szybkością działania.

Przy ograniczeniu zastosowania systemu do reprezentacji obiektów leżących w jednej płaszczyźnie (np. powierzchnia ziemi) lub wydzieleniu w systemie oddzielnych torów obiektów trójwymiarowych, jest ponadto możliwe uniezależnienie wypełniania całej płaszczyzny ekranu w torze odległości od kształtu wyświetlanych obiektów i wypełnianie planu odległości jedynie na podstawie informacji o położeniu obserwatora. W przeciwnym razie do wypełniania planu odległości należy zastosować układy generujące (na podstawie współrzędnych wielokąta i położenia obserwatora) informację o odległości dla każdego piksela wielokąta.

Przy założonej rozdzielczości obrazu rzeczywista częstotliwość przebiegu synchronizującego wyświetlanie pikseli wynosi 56 MHz, co daje czas generowania piksela ok. 18 ns. Ponieważ jednocześnie częstotliwość wyświetlania kodów obrazu, wynosząca 60 Hz, przewyższa dwukrotnie założoną częstotliwość ich generowania można zastosować buforowanie, umożliwiające dwukrotne zmniejszenie szybkości generowania strumienia pikseli (do 36 ns/piksel). Do realizacji operacji arytmetycznych można zastosować układy szybkich pamięci RAM, do których uprzednio zostaną wpisane dane stanowiące stabilizowane wyniki funkcji wykorzystywanych do obliczeń. Takie rozwiązanie pozwala również wprowadzić odległość l_m jako parametr zmieniający się co kadr obrazu i uzyskać efekt zmiany przezroczystości ośrodka.

Główne zalety przedstawionego rozwiązania to oszczędność pamięci (w planie atrybutów można przechowywać tylko indeks do tablicy pośredniej kolorów wyjściowych, mieszczący się na 8–12 bitach) oraz prostota struktury i duża szybkość wypełniania planu atrybutów dzięki wypełnianiu wielokątów stałym kolorem.

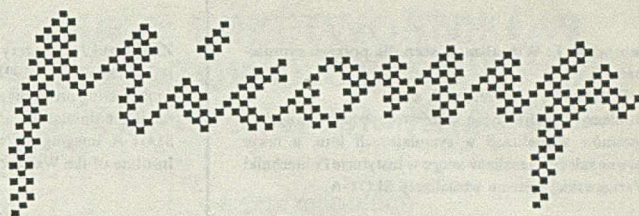
★ ★ ★

Przedstawiony schemat obrazowania zmiany koloru został zasymulowany, a obecnie trwają prace projektowe nad stroną układową urządzenia, zwłaszcza nad układem wypełniania pamięci ekranu w torze odległości. Prezentowane dwa tory graficzne stanowią jedynie część projektowanego systemu graficznego. Planuje się rozbudowę urządzenia o dalsze tory, których obecność umożliwi reprezentację innych zjawisk i efektów, np. tekstury lub niejednorodnego oświetlenia.

LITERATURA

- [1] Fujimoto A., Perrot C.G., Iwata K.: A 3D Graphics Display Processor with Depth Buffer and Pipeline Processor. IEEE Computer Graphics and Applications, No. 6, pp. 11–23, 1984
- [2] Rzeszut J.: Efekt perspektywy powietrznej w generowaniu obrazu dla symulatorów. INFORMATYKA, nr 3/1991.

Informujemy prenumeratorów INFORMATYKI, że w związku z podwyższeniem ceny czasopisma do 12 tys. zł za egzemplarz (cena ulgowa 8400 zł) Dział Kolportażu zwróci się do Państwa pisemnie, określając kwotę, którą należy dopłacić do już uiszczonej prenumeraty. Aktualne ceny prenumeraty podajemy na II stronie okładki.



ZAKŁAD SYSTEMÓW MIKROKOMPUTEROWYCH **MiCOMP**

40-045 Katowice, ul. Astrów 7
telefon 513-086, telefaks 518-628,
teleks 315687

Systemy teletransmisji danych
ICL/ODRA

Integracja systemów ICL i ODRA
z mikrokomputerami standardu PC
i systemami NOVELL, UNIX SCO

procesor sieci rozległej WAN
MiCOMP 8075
(emulacja ICL 7503)

most NOVELL – ICL/ODRA
NetCROSS-FXBM
program teletransmisji danych
MiCROS-FXBM (PC, rozproszone bazy danych,
wersja sieciowa – NOVELL),

skaner **MPXSCAN-8007**,

adaptory, modemy, testery,
emulatory terminali

karty BSC do połączeń komputerów:
PC/XT/AT z PC/XT/AT
PC/XT/AT z IBM/370, R32, R34
z emulatorami IBM 3270/3780
PC/XT/AT z ICL

karty kompatybilne z **IRMA** do połączeń
PC/XT/AT z IBM 370
lokalnie przez IBM 3274

karty **SNA/SDLC**

konwertery o szybkościach do 19200 bps
do zdalnego przyłączania terminali

przystawki teleksowe współpracujące
z mikrokomputerami PC/XT/AT

Instalacje w największych systemach
na terenie całego kraju!

0/23/90

<p>Zabrodzki J.: Wizualizacja scen dla potrzeb symulatorów lotu INFORMATYKA 1991, nr 3, s. 1 Problemy konstrukcyjne oraz stosowane rozwiązania systemów wizualizacji w symulatorach lotu, a także główne założenia realizowanego w Instytucie Politechniki Warszawskiej systemu wizualizacji SLOT-A.</p>	<p>Zabrodzki J.: Imagery for flight simulators INFORMATYKA 1991, No. 3, p. 1 Constructing problems and applied solutions of imagery for flight simulators, as well as main principles of the SLOT-A imaging system realized in the Informatics Institute of the Warsaw Technical University.</p>	<p>Zabrodzki J.: Szenenabbildung für Flugsimulatoren INFORMATYKA 1991, Nr. 3, S. 1 Konstruktionsprobleme und angewendete Lösungen von Abbildungssystemen in Flugsimulatoren, sowie wichtigsten Voraussetzungen des im Institut für Informatik der Warschauer Technischen Universität realisierten SLOT-A-Abbildungssystems.</p>
<p>Chrzyszcz J.: Struktura wieloprocessorowa do celów cyfrowego generowania obrazu INFORMATYKA 1991, nr 3, s. 5 Konceptja architektury oraz funkcjonowania wieloprocessorowego systemu graficznego przeznaczonego do wizualizacji w symulatorach lotu.</p>	<p>Chrzyszcz J.: Multiprocessor structure for image digital generation INFORMATYKA 1991, No. 3, p. 5 Architecture and operation conception of multiprocessor graphical system for imaging in flight simulators.</p>	<p>Chrzyszcz J.: Multiprozessorstruktur für Zifferngenerierung eines Bildes INFORMATYKA 1991, Nr. 3, S. 5 Architekturkonzeption und Funktionierung eines Multiprozessorgrafiksystems für Abbildung in Flugsimulatoren.</p>
<p>Pajak A.: SDeL – język opisu scen geometrycznych INFORMATYKA 1991, nr 3, s. 8 Charakterystyka języka SDeL przeznaczonego do tworzenia baz danych geometrycznych w komputerowej symulacji lotu.</p>	<p>Pajak A.: SDeL – a language for geometric images description INFORMATYKA 1991, No. 3, p. 8 Characteristics of the SDeL language for geometric databases building in computerized flight simulators.</p>	<p>Pajak A.: SDeL – eine Sprache für Beschreibung der geometrischen Szenen INFORMATYKA 1991, Nr. 3, S. 8 Eine Charakteristik der SDeL-Sprache, die für Geometriedatenbankschaffung in computerisierten Flugsimulation bestimmt ist.</p>
<p>Chrzyszcz J.: Układ szybkiego wypełniania pamięci obrazu INFORMATYKA 1991, nr 3, s. 11 Konceptja i sposób implementacji sterownika do szybkiego wypełniania pamięci obrazu w systemie wizualizacji lotu.</p>	<p>Chrzyszcz J.: Circuit for high-speed filling of image memory INFORMATYKA 1991, No. 3, p. 11 Conception and implementation method of the controller for high-speed filling of image memory in flight simulator's imaging system.</p>	<p>Chrzyszcz J.: Schaltung für schnelles Ausfüllen des Bildspeichers INFORMATYKA 1991, Nr. 3, S. 11 Konzeption und Implementierungsmethode einer Steueranlage für schnelles Ausfüllen des Bildspeichers in einem Abbildungssystem eines Flugsimulators.</p>
<p>Rzeszut J.: Algorytm rysowania barwnego odcinka i jego wykorzystanie do cieniowania wnętrza trójkąta INFORMATYKA 1991, nr 3, s. 14 Charakterystyka algorytmu generowania obrazu obiektów trójwymiarowych w symulatorach lotu.</p>	<p>Rzeszut J.: Algorithm for color section drawing and its application for triangle area shading INFORMATYKA 1991, No. 3, p. 14 Characteristics of the algorithm for image generation of 3D objects in flight simulators.</p>	<p>Rzeszut J.: Algorithmus für Zeichnen eines Farbabschnitts und seine Ausnützung zur Schattierung eines Dreieckinneren INFORMATYKA 1991, Nr. 3, S. 14 Eine Charakteristik des Algorithmus für Generierung des Bildes von dreidimensionalen Objekten in Flugsimulatoren.</p>
<p>Prokop, Stępień C.: Sposób odwzorowania światła na ekranie monitora graficznego w animowanych scenach nocnych. INFORMATYKA 1991, nr 3, s. 17 Sprzętowe rozwiązanie problemu odwzorowania światła na ekranie przy animacji scen nocnych w symulatorach lotu.</p>	<p>Prokop T., Stępień C.: Method for lights repatterning on graphic display in animated night images INFORMATYKA 1991, No. 3, p. 17 Hardware solution of light repatterning problem on display in animated images in flight simulators.</p>	<p>Prokop T., Stępień C.: Methode für Lichterabbildung auf Bildschirm eines Grafikmonitors in animierten Nachtszenen INFORMATYKA 1991, Nr. 3, S. 17 Hardwarelösung des Problems von Lichterabbildung auf Bildschirm bei Animierung der Nachtszenen in Flugsimulatoren.</p>
<p>Rzeszut J.: Efekt perspektywy powietrznej w generowaniu obrazu dla symulatorów INFORMATYKA 1991, nr 3, s. 19 Charakterystyka zjawiska perspektywy powietrznej i jej zastosowania w symulatorach lotu.</p>	<p>Rzeszut J.: Aeronautic perspective effect in image generation for simulators INFORMATYKA 1991, No. 3, p. 19 Characteristics of aeronautic perspective and its application in flight simulators.</p>	<p>Rzeszut J.: Lufttraumperspektiveeffekt in Bildgenerierung für Flugsimulatoren INFORMATYKA 1991, Nr. 3, S. 19 Eine Charakteristik der Lufttraumperspektiveerscheinung und ihrer Anwendung in Flugsimulatoren.</p>
<p>Mazur G.B.: Realizacja efektu zmiany koloru obiektu w ośrodku optycznym w systemie generowania obrazu w czasie rzeczywistym INFORMATYKA 1991, nr 3, s. 22 Sprzętowe rozwiązanie zmiany koloru obiektu przy generowaniu obrazu na ekranie w czasie rzeczywistym.</p>	<p>Mazur G.B.: Realization of object's color change in optical medium for real-time image generation system INFORMATYKA 1991, No. 3, p. 22 Hardware solution of object's color change in real-time image generation on display.</p>	<p>Mazur G. B.: Realisierung des Effektes von Objektsfarbeänderung in optischen Umwelt eines Echtzeitsystems für Bildgenerierung INFORMATYKA 1991, Nr. 3, S. 22 Hardwarelösung der Objektsfarbeänderung bei Echtzeitgenerierung auf einem Bildschirm.</p>

Koło Zainteresowań Cybernetycznych
działające przy
Wydziale Cybernetyki
Wojskowej Akademii Technicznej
w Warszawie

**zawiadamia że seminarium
studenckich kół naukowych
informatyki**

INFOSEM-91

organizowane przy współpracy z Przedsiębiorstwem Projektowo-Wdrożeniowym AWAT sp. z o.o. odbędzie się w dniach **25-28 kwietnia 1991 r.** na terenie Wojskowego Domu Wypoczynkowego w Ryni nad Zalewem Żegrzyńskim. Bliższych informacji na temat zasad uczestnictwa i przygotowania referatów udzielają opiekunowie Koła:

dr inż. Krzysztof Liderman tel. 36-97-01

mgr inż. Marek Deutsch tel. 36-06-61 w. 3128

KARTY PRZETWORNIKÓW ANALOGOWO-CYFROWYCH do komputerów IBM XT/AT/386

na podzespołach

ANALOG DEVICES, MAXIM BURR, BROWN

- | | | |
|--|--|--|
| <input type="checkbox"/> 12, 14, 22 bity | <input type="checkbox"/> praca w tle | <input type="checkbox"/> izolacja galwaniczna |
| <input type="checkbox"/> pamięć buforowa | <input type="checkbox"/> wzmacniacze pomiarowe | <input type="checkbox"/> jednoczesne próbkowanie |

próbkowanie do 1 MHz!!!

KARTY PROCESORÓW SYGNAŁOWYCH

TMS320C25 MOTOROLA 56001 AT&T DSP32C

Bogate oprogramowanie

Kompleksowa obsługa zamówień

Serwis

Gwarancja

Pomoc w instalacji

PRACOWNIA USŁUG INFORMATYCZNYCH I KOMPUTEROWYCH 'CONVERT'

50-412 Wrocław, ul. Mazowiecka 17,
telefon 300-11 w. 210

0/3/91

FUNDACJA POMOCY MATEMATYKOM i INFORMATYKOM NIEPEŁNOSPRAWNYM RUCHOWO - WARSZAWA

00-901 Warszawa, PKiN, skr. poczt. 22
telefaks 200144, teleks 813556, tel. 201778, 200211
wew. 2464, 2529

Szanowni Państwo!

Uprzejmie prosimy o udzielanie pomocy naszej Fundacji. Zadaniem jej jest pomoc niepełnosprawnym ruchowo kolegom (w większości korzystającym z wózków inwalidzkich) pracującym lub uczącym się w województwie warszawskim.

Cel Fundacji realizujemy przez:

- ☐ umożliwienie podopiecznym działalności zawodowej na komputerach osobistych;
- ☐ pomoc materialną, socjalną i prawną.

Najpilniejszą potrzebą Fundacji jest uzyskanie sprzętu komputerowego dla naszych podopiecznych. Będziemy również zobowiązani za okazanie pomocy w każdej formie. Prosimy też dopomóc w skontaktowaniu się z nami osób, które chciałyby korzystać z opieki Fundacji.

Z poważaniem

Wiceprzewodniczący Zarządu Fundacji
dr W. Nowiński

Wpłaty na konto Fundacji można dokonywać w:

- Banku Przemysłowo-Handlowym: BPH XIV O/Warszawa, 320007-3766-132-3,
- kasie Fundacji: Pałac Kultury i Nauki, X piętro, pokój 1021, godz. 9-15,
- z zagranicy na konto: Kontinent Nr 3 150 000, Deutsche Bank Bonn, BLZ 380 700 59 Stichwort BEHINDERTE/POLEN.

PRZEDSIĘBIORSTWO ZASTOSOWAŃ INFORMATYKI

meditronik

oferuje

UM7106	18 000
UM3481,2,3,4,10,11	8200
UM66T	3600
UM82450	14 000
UM8250B	14 000
UM8253-5	16 500
UM8259A-2	27 300
UM82C11	13 000
UM82C55A	16 500
UM82C8167	19 500
UM6116U-35(ns)	25 500
UM6116-2L	11 500
UM6264-10	20 000
UM62256A-10L	52 000

Nasz adres:

PZI „MEDITRONIK”

00-194 Warszawa, ul. Dzika 4

telefon 635-22-63, 635-22-64

telefaks 635-21-95

teleks 816075 medi pl

0/24/90

POZNAJ SIŁĘ NASZYCH SYSTEMÓW INFORMATYCZNYCH!

JUNISOFTEx Sp. z o.o.

44-100 Gliwice ul. Konstytucji 11

Tel. 31-75-10, 31-90-81 do 88 wewn. 250, 272, teleks 036233

Już od 3 lat oferuje kompleksową komputeryzację przedsiębiorstw, a zwłaszcza:

- profesjonalne nowoczesne systemy informatyczne oparte na kompilatorze języka Turbo Pascal 5.5 firmy BORLAND (typu multiusers), pracujące w sieciach NOVELL 2.15 i 3.1 i wykorzystujące interfejs BTRIEV amerykańskiej firmy SoftCraft – najefektywniejszy i najszybszy system obsługi dużych i bardzo dużych baz danych (do 4 miliardów rekordów w pojedynczej bazie danych).

Polecamy szczególnie systemy zintegrowane

- wielodostępny system finansowo-księgowy wg trójstopniowej analityki w podstawowym układzie: xxx-xxxx-xxxx-xxxxx z możliwością graficznej prezentacji wyników,
- wielodostępny system gospodarki materiałowej wykorzystujący w strukturze indeksowej SWW i KTM z automatycznym wyliczaniem cyfry kontrolnej (długość indeksu materiałowego 13 + 3 cyfry),
- wielodostępny system gospodarki wyrobami gotowymi wykorzystujący w strukturze indeksowej SWW i KTM,
- wielodostępny system gospodarki przedmiotami nietrwałymi wykorzystujący w strukturze indeksowej SWW i KTM,
- wielodostępny system środków trwałych wykorzystujący klasyfikację środków trwałych GUS,
- wielodostępny system fakturowania sprzedaży,
- wielodostępny system kadrowo-płacowy.

Wszystkie oferowane systemy są zintegrowane ze sobą zgodnie z zasadami rachunkowości księgowej przedstawionymi w miesięczniku RACHUNKOWOŚĆ nr 9, 1990.

Oferujemy również:

- sprzęt komputerowy (sprawdzony w eksploatacji) firm:
 - ALR (Advanced Logic Research) i Storage Dimensions z USA,
 - AZTECH, WEARNES TECHNOLOGY, NAC i inne z Singapuru,
- dostawę i instalację licencjonowanej sieci Novell 2.15 i 3.1,
- usługi w zakresie opracowywania zakładowych indeksów materiałowych wg klasyfikacji GUS (SWW i KTM).

Organizujemy:

- szkolenia i prezentację systemów u naszych kontrahentów.

Gwarantujemy:

- najwyższą jakość naszych systemów i usług,
- najszybsze w Polsce wdrożenia systemów (od 1 do 3 miesięcy). W przypadku niedotrzymania przez nas terminów zwracamy 25% poniesionych nakładów za każdy miesiąc zwłoki. Dla klientów zamawiających kompleksowe usługi – znaczne zniżki cenowe.

Poszukujemy niezawodnych

- pośredników i dealer'ów naszych systemów informatycznych (oferujemy bardzo atrakcyjne warunki współpracy),
- dostawców i pośredników dobrego i taniego sprzętu komputerowego.

Z siły naszych systemów skorzystały dotąd następujące firmy:

Huta ŁAZISKA w Łaziskach Górnych
Walcownie Metali w Czechowicach-Dziedzicach
Centrala Zaopatrzenia Górnictwa w Bytomiu
Mikołowska Fabryka Transformatorów w Mikołowie
BPIKD PREFAMET-ZREMB w Gliwicach
BPIKD METALCHEM w Gliwicach
Instytut Informatyki Teoretycznej i Stosowanej PAN w Gliwicach
Instytut Spawalnictwa w Gliwicach
PP ENERGOROZRUCH w Gliwicach
Fabryka palenisk mechanicznych w Mikołowie
Zakłady Mechaniczne Górnictwa i Energetyki WIROMET w Mikołowie
Przędzalnia Bawełny PRZYJAŻŃ w Zawierciu
PRT TERMOIZOLACJA w Zabrze
Przedsiębiorstwo Usług Informatycznych ETOM w Mikołowie
Zakład Informatyki i Techniki Komputerowej MEGA w Gliwicach
Mikołowska Fabryka Maszyn Górniczych MIFAMA w Mikołowie
Zakłady Produkcji Urządzeń Mechanicznych ELWO w Pszczynie
Zakłady Sprzętu Sieciowego i Elektroinstalacyjnego POLAM w Katowicach
Urząd Miasta w Mikołowie

tel. 24-14-06
tel. 52-35-1
tel. 81-92-91
tel. 26-25-41
tel. 38-20-51
tel. 31-56-33
tel. 31-71-50
tel. 31-00-11
tel. 37-66-25
tel. 26-20-02
tel. 26-23-46
tel. 22-39
tel. 71-40-21
tel. 57-10-50
tel. 31-52-42
tel. 26-22-66
tel. 30-61
tel. 52-80-31
tel. 26-23-90

Chcesz mieć pełną gwarancję i niezawodne efekty komputeryzacji swojego przedsiębiorstwa – współpracuj z firmą JUNISOFTEx!