

Stawomir Jasiński, Marek Malarski, Jacek Skorupski  
Politechnika Warszawska

ANALIZA NUMERYCZNEJ METODY ZOBRAZOWANIA WYBRANYCH ELEMENTÓW  
LOTNISKA DLA SYMULATORA TWR

METHOD FOR TWR SIMULATOR ANALYSIS OF CHOSEN ELEMENTS OF AIRPORT  
AREA VISUALISATION

АНАЛИЗ ЧИСЛЕННОГО МЕТОДА ОТОБРАЖЕНИЯ ИЗБРАННЫХ ЭЛЕМЕНТОВ  
АЭРОДРОМА ДЛЯ ТРЕНАЖЕРА ТWR

**Streszczenie:** W artykule opisano metodę umożliwiającą opis złożonych obiektów takich jak samolot w celu przeprowadzenia ich animacji na ekranie komputera. Opracowano implementację programową umożliwiającą wprowadzanie danych opisujących poszczególne części samolotu i obiektów lotniskowych zgodnie z opracowaną strukturą.

**Summary:** In this paper a method of describing complex elements (such as airplane) used for their computer animation is presented. A computer program for data input (describing parts of airplanes and airport facilities according to given structure) is elaborated.

**Резюме:** В статье представлен метод позволяющий описывать сложные объекты типа самолет с целью их анимации на экране компьютера. Показана программа для ввода, согласно с разработанной структурой, данных описывающих отдельные части самолета и объекты аэродрома.

## 1. Wstęp

Przy rozwiązywaniu wielu zagadnień dotyczących projektowania i eksploatacji systemów, których koszt budowy jest bardzo duży oraz którym stawiane są wysokie wymagania dotyczące bezpieczeństwa, stosuje się coraz częściej metody symulacji. W tym celu używane są symulatory, które można podzielić na dwie grupy:

Symulatory badawcze - umożliwiają uzyskanie przez człowieka informacji o działaniu złożonego systemu w skomplikowanych sytuacjach. Dotyczy to zwłaszcza konstrukcji nowych systemów, które mogą być badane równoległe z ich projektowaniem lub budową prototypów [6]. Symulator umożliwia dobór optymalnych parametrów układów sterowania, analizę własności dynamicznych itp. W lotnictwie można je ponadto stosować do opracowania nowych technik pilotażu, systemów sterowania [2], analizy przyczyn wypadków lotniczych [4] itp.

Symulatory treningowe - stosowane są do celów szkoleniowych. Umożliwiają one opanowanie przez operatora umiejętności manualnych, nawyków i odruchów niezbędnych dla sprawnego wykonywania zadania zarówno w sytuacjach typowych jak i przypadkach szczególnych: awarie, uszkodzenia itp. Korzyści ze stosowania symulatorów wynikają często z niższych kosztów eksploatacji. Niekiedy jednak cena symulatora jest wyższa od ceny urządzenia, co jest jednak rekompensowane poprzez zmniejszenie ryzyka popełnienia błędu w czasie pracy przy użyciu właściwego urządzenia. Dochodzą do tego efekty niewymierne w postaci pełnego bezpieczeństwa szkolenia podstawowego, treningu i prób, możliwości kontroli, zapisu i analizy aktualnej sytuacji [12].

Służby kontroli ruchu lotniczego zapewniają bezpieczny, zgodny z planem i sprawny przelot statków powietrznych. Wykonywanie tych zadań odbywa się przy użyciu wielu elementów wzajemnie powiązanych i zmiennych zależnie od czasu, miejsca i warunków atmosferycznych. Różnorodność zadań wykonywanych przez lotnictwo uniemożliwia utworzenie jednolitej i powszechnej metodyki sterowania całym ruchem lotniczym, nawet w ramach przestrzeni powietrznej jednego państwa. W obszarach o dużym natężeniu ruchu lotniczego konieczne jest zapewnienie systemu scentralizowanej kontroli sytuacji w powietrzu (systemu ATC), umożliwiającego rozwiązywanie sytuacji konfliktowych. System ten powinien zapewnić możliwość

spełnienia wymagań planów lotów oraz zapewnienia bezpiecznego, płynnego i bezkolizyjnego ruchu statków powietrznych. Zadania służby ruchu lotniczego obejmują m.in.: zapobieganie kolizjom statków powietrznych podczas lotu, zapobieganie kolizjom statków powietrznych; przeszkodami na polu manewrowym lotniska, usprawnianie i utrzymanie uporządkowanego przepływu ruchu lotniczego. Służba kontroli ruchu lotniczego dzieli się na: służbę kontrol obszaru ACC (Area Control Centre), służbę kontroli zbliżania APP (Approach Control), służbę kontroli lotniska TWR (Tower),

Strefa kontrolowana lotniska obejmuje płytę lotniska i przestrzeń powietrzną rozciągającą się od powierzchni ziemi do wysokości dolnej granicy drogi lotniczej lub rejonu kontrolowanego węzła lotnisk nad nią leżących. W granicach strefy kontrolowanej lotniska zapewnione jest wykonywanie operacji: oczekiwania, podejścia do lądowania, startu i nabierania wysokości, manewru nieudanego podejścia. Ruch lotniskowy obejmuje ponadto wszelki ruch na polu manewrowym lotniska ruch wszystkich statków powietrznych wykonujących loty w pobliżu lotniska. Zadanie służby kontroli lotniska TWR polega na takim przestrzennym rozmieszczeniu statków powietrznych względem siebie i względem terenu, aby:

- statkom nie groziło niebezpieczeństwo kolizji w powietrzu i na polu manewrowym lotniska,
- każdy ze statków miał możliwie najbardziej dogodny warunki lotu,
- ruch lotniczy odbywał się możliwie płynnie bez zbędnych oczekiwań, zmian poziomu lotu itp.

Jednym ze sposobów pomagających w wykonaniu zadań ATC jest zapewnienie między statkami powietrznymi separacji. Separacje dzielą się na: wzdłużne, boczne i pionowe, a są one normami odległości zapewniającymi bezpieczeństwo.

Kontroler ruchu lotniczego ma zapewniać bezpieczny, zgodny z planem i sprawny przelot statków powietrznych. Poprawne wykonywanie tych zadań przez człowieka zależy od takich czynników, jak: predyspozycje, kwalifikacje, praktyka, samopoczucie i wiele innych. Prawidłowe szkolenie i przygotowanie do pracy kontrolera wymaga treningu przy użyciu symulatora, dającego mu możliwość poznania czynności, procedur i zasad zachowania w określonych sytuacjach podczas sterowania ruchem lotniczym. Ma umożliwić kontrolerowi zapoznanie się z procedurami stosowanymi w określonych sytuacjach oraz nabycie pewnych nawyków poprzez wielokrotne, często automatyczne wykonywanie określonych czynności zależnie od sytuacji [11]. Symulator wieży kontrolnej (TWR) zapewniający wizualizację aktualnej sytuacji na płycie lotniska i w powietrzu powinien umożliwić lepsze przygotowanie kontrolerów. Wyniki wstępnych badań nad konstrukcją oprogramowania takiego symulatora przedstawiono w tym artykule. Celem omawianej w tym artykule pracy jest minimalizacja złożoności opisu obiektu (samolotu, lotniska) przy założonym stopniu podobieństwa do pierwowzoru.

## 2. Wymagania stawiane układowi wizualizacji

Z badań nad sposobem odbierania przez człowieka informacji o otaczającym świecie wynika, że przeważająca ich ilość dociera w postaci wizualnej (szacunkowo, informacja uzyskana za pośrednictwem wzroku stanowi 80% informacji zdobytej przez wszystkie zmysły). Zapewnienie realizmu symulowanej sytuacji wymaga więc przede wszystkim odtworzenia bodźców wzrokowych dochodzących do operatora symulowanego procesu.

Różnorodność systemów wizualizacji wymaga określenia podstaw teoretycznych procesów tworzenia obrazu, sposobu odbioru wrażeń wzrokowych przez człowieka, a także wymagań stawianych układowi wizualizacji. Na proces widzenia składa się suma wrażeń odbieranych na siatkówkach oczu, następnie przekazywanych do ośrodka widzenia w mózgu. Badania nad działaniem oka i psychologią widzenia wskazują na fakt, że obraz w mózgu nie jest prostą kopią obrazu na siatkówce. Przynajmniej część analizy obrazu odbywa się już wewnątrz oka. Mózg człowieka odzwierciedla obiekty ze świata zewnętrznego, opierając się na znajomości (świadomej lub nie, przeważnie opartej na doświadczeniu) odpowiednich wzorców (modeli). Zadaniem sztucznego generowania obrazu jest oprócz pełnego przestrzegania praw jego

tworzenia, wykorzystanie ich także do wprowadzania celowych złudzeń. Odpowiednie graficzne środki wyrazu powinny umożliwić mózgowi człowieka odtworzenie takiego otoczenia przestrzennego, do jakiego jest przyzwyczajony [8]. Wymagania te mogą odnosić się do różnych cech tworzonego obrazu. Część wymagań dotyczy jakości generowanego obrazu, którą można oceniać ze względu na:

- realizm odwzorowania sytuacji np. odwzorowanie cech terenu,
- wierność odtworzenia obrazu obiektów istniejących w rzeczywistości.

Realizm i łatwa adaptacja wiążą się z możliwością dostosowania symulatora do dowolnego portu lotniczego oraz przedstawienia szczególnych dla danego portu ścieżek podejścia, co umożliwi każdemu kontrolerowi odbycie treningu w warunkach bardzo zbliżonych do tych, jakie spotykane są na jego stanowisku pracy. Realizm wiąże się również z możliwością przedstawienia różnych typów samolotów, helikopterów i pojazdów w taki sposób, aby można było je z łatwością rozpoznawać i odróżniać. Symulator powinien umożliwiać symulowanie różnych sytuacji na ziemi i w powietrzu, zarówno o małym jak i dużym stopniu komplikacji, zależnie od doświadczenia osoby z niego korzystającej.

Inne wymagania odnoszą się do zdolności sterowania obrazem, która zależy od:

- liczby stopni swobody przemieszczeń obserwatora względem terenu,
- zakresu przemieszczeń obserwatora (wielkości obszaru), którego obraz może być wytworzony,
- dynamiki zmian obrazu - szybkości reakcji na przemieszczenia obrazu,
- powtarzalności generowanego obrazu,
- występowanie w obrazie obiektów ruchomych (np. statków powietrznych, samolotów).

W układach cyfrowej generacji obrazu wykorzystywany jest jedynie matematyczny opis przestrzeni, której obraz należy wytworzyć. Brak jakichkolwiek fizycznych odpowiedników (modeli) terenu powoduje, iż system taki może być bardzo uniwersalny i elastyczny. Wszelkiego rodzaju zmiany mogą być dokonywane poprzez wymianę informacji zawartej w pamięci. Szczegóły procesów zachodzących w układzie cyfrowej generacji obrazu zależą od przeznaczenia modelu, jego możliwości, charakterystyki syntezy obrazu, postaci danych i rodzaju urządzenia zewnętrznego.

W grafice komputerowej zależność między metodami rozwiązywania zadań a sprzętem, na którym się je realizuje, jest wyjątkowo silna [1]. Podczas projektowania symulatora bardzo ważnym elementem jest zapewnienie właściwej wizualizacji problemu. Istotny dla sposobu tworzenia rysunku jest tryb pracy monitora graficznego.

### 3. Geometria w przestrzeni

Punkty w przestrzeni określa się podając ich współrzędne w określonym układzie. Na ogół jest to układ kartezjański, ale czasami może być wygodniejszy, np. układ współrzędnych sferycznych. W grafice komputerowej bardzo ważna jest umiejętność nie tylko przekształcania danego obiektu, lecz także opisywania go w różnych układach współrzędnych. Na przekształcanie punktów w przestrzeni trójwymiarowej składają się: przesunięcie, skalowanie, obrót wokół osi układu, rzutowanie, przekształcenie układu danych do układu obserwatora.

**PRZESUNIĘCIE** (translacja) o wektor  $[tx, ty, tz]$  punktu  $P=(x, y, z)$  daje punkt  $P'=(x', y', z')$  o współrzędnych:  $x'=x+tx$ ,  $y'=y+ty$ ,  $z'=z+tz$ .

**SKALOWANIE** definiuje się zależnościami:  $x' = sx \cdot x$ ,  $y' = sy \cdot y$ ,  $z' = sz \cdot z$ .

**OBRÓT WOKÓŁ OSI UKŁADU** - jakó podstawowy przyjmuje się lewoskrętny układ współrzędnych. Dodatni zwrot kąta obrotu wokół danej osi to zwrot przeciwny do kierunku ruchu wskazówek zegara.

Obrazem punktu  $P=(x, y, z)$  po obrocie o kąt  $\omega$  wokół osi  $x$  jest punkt  $P'=(x', y', z')$ , gdzie:  $x' = x$ ,  $y' = y \cos \omega - z \sin \omega$ ,  $z' = y \sin \omega + z \cos \omega$ .

Obracając punkt  $P$  o kąt  $\omega$  wokół osi  $y$  otrzymujemy punkt  $P'=(x', y', z')$  o współrzędnych:  $x' = z \sin \omega + x \cos \omega$ ,  $y' = y$ ,  $z' = z \cos \omega - x \sin \omega$ ,

Wreszcie, przy obrocie o kąt  $\omega$  wokół osi z punkt P jest przekształcany w  $P'=(x',y',z')$  o współrzędnych:  $x' = x \cos\omega - y \sin\omega$ ,  $y' = x \sin\omega + y \cos\omega$ ,  $z' = z$

**RZUTOWANIE** - jest to przekształcenie podstawowe w grafice trójwymiarowej, gdyż komputerowa wizualizacja jakiegokolwiek obiektu przestrzennego wymaga odwzorowania go na płaski ekran monitora. Metody rzutowania można podzielić na dwie klasy. Pierwszą stanowią rzuty równoległe, drugą rzuty perspektywiczne. Rzut równoległy zachowuje równoległość prostych, stosunek długości odcinków równoległych i związki miarowe figury płaskiej równoległej do płaszczyzny rzutowania. Rzut perspektywiczny pozwala na bardziej realistyczną wizualizację obiektów trójwymiarowych i daje wrażenie głębi. W rzucie perspektywicznym wszystkie proste (promienie rzutowania) mają wspólny punkt, nazywany środkiem rzutowania - rzutnią. Odległość punktu od rzutni decyduje o "deformacji" rysunku. Znalazienie współrzędnych rzutu  $P'$  punktu P sprowadza się do wyznaczenia przecięcia płaszczyzny  $\Pi$  i prostej (promienia rzutowania) określonej punktem P i kierunkiem  $k$  przy rzucie równoległym lub punktem P i środkiem rzutowania E w przypadku rzutu perspektywicznego.

**PRZEKSZTAŁCENIE UKŁADU DANYCH DO UKŁADU OBSERWATORA** - rzutowanie daje współrzędne punktu P' w układzie trójwymiarowym, w którym został opisany rysowany obiekt (układzie danych). Do wizualizacji są potrzebne współrzędne w układzie dwuwymiarowym określonym na płaszczyźnie rzutowania  $\Pi$ . Takie współrzędne można wyznaczyć stosując metodę przekształcania układu danych do układu obserwatora. Układem obserwatora nazywamy układ współrzędnych  $\underline{O} \ x \ y \ z$ , w którym płaszczyzna rzutowania  $\Pi$  pokrywa się z płaszczyzną  $z=0$ . W przypadku rzutowania perspektywicznego utożsamiamy położenie obserwatora ze środkiem rzutowania E o współrzędnych  $(x_E, y_E, z_E)$  w wyjściowym układzie danych  $Oxyz$ . Przyjmujemy, że płaszczyzna rzutowania  $\Pi$  jest prostopadła do OE.

Układ obserwatora nie jest określony jednoznacznie, możemy przyjąć, że jego początek  $\underline{O}$  pokrywa się z początkiem  $\underline{O}$  układu danych. Przekształcenie układu danych do układu obserwatora polega na wykonaniu takich obrotów wokół osi układu, by wektor  $n=[x_n, y_n, z_n]$  miał kierunek osi  $z$  nowego układu i przeciwny do niej zwrot. Szukaną transformację można otrzymać w dwóch krokach:

1. Wykonujemy obrót  $R(\omega, z)$  układu osi z o kąt  $\omega = \arctg(\frac{y_n}{x_n})$  dla  $x_n \neq 0$  lub  $\omega = 90^\circ$ , gdy  $x_n = 0$ . W nowym układzie  $Ox'y'z'$  wektor  $[x_n, y_n, z_n]$  będzie miał składowe  $[s, 0, z_n]$ , gdzie  $s = (x_n^2 + y_n^2)^{0.5}$ .

2. Obracamy  $R(\psi, y')$  układ  $Ox'y'z'$  o kąt  $\psi = \arctg(\frac{z_n}{s}) + 180^\circ$  wokół osi  $y'$ . Składowe wektora  $[s, 0, z_n]$  w otrzymanym układzie  $Ox''y''z''$  są równe  $(0, 0, t)$ , gdzie  $t = (s^2 + z_n^2)^{0.5}$ .

Złożenie przekształceń  $R = R(\omega, z)R(\psi, y')$  przeprowadza układ danych do układu obserwatora. Takie rozwiązanie nie jest jednak wystarczające. Rzutnia  $\Pi$  jest wprawdzie prostopadła do nowej osi  $z''$ , ale pozostałe osie  $x''$  i  $y''$  mogą mieć dowolne położenie. Wprowadzone wcześniej dwa obroty można uzupełnić jeszcze jednym, dobranym np. tak, by "trzymać pion": - wersor  $e_x = [0, 1, 0]$  wyznaczający pion w układzie danych powinien mieć w nowym układzie współrzędnych składową  $x$  równą zeru. Obracamy  $R(\eta, z'')$  układ o kąt  $\eta =$

$-\arctg(\frac{r_{21}}{r_{22}})$  wokół osi  $z''$ , na podstawie składowych wersora  $e_x$  w układzie przekształconym obrotami:  $[r_{21}, r_{22}, r_{23}]$ . Taka transformacja, zmienia jedynie osie  $x''$  i  $y''$  na rzutni  $\Pi$ , zachowując prostopadłość do tej płaszczyzny osi  $z''$ .

Ostatecznie, przejście od układu danych do układu obserwatora dane jest złożeniem:

$$R(\omega, z)R(\psi, y')R(\eta, z'')$$

#### 4. Struktura danych układu wizualizacji

Artykuł prezentuje wyniki wstępnego etapu prac mającego na celu opracowanie struktury danych, dającej możliwość opisu złożonych obiektów takich, jak: samolot, radar itp. w celu przeprowadzenia ich animacji na ekranie komputera. Na bazie przeprowadzonej analizy opracowano program umożliwiający wprowadzanie danych opisujących poszczególne części samolotu i obiektów lotniskowych, zgodnie z opracowaną strukturą, a następnie takie przetwarzanie danych, aby uzyskać rzuty prostokątne poszczególnych części wzdłuż osi  $x$ ,  $y$  i  $z$ , które zapewniają lepsze warunki do wprowadzania poprawek podczas opisywania obiektu, niż rzut perspektywiczny. Program miał zapewnić możliwość połączenia ze sobą poszczególnych części w celu otrzymania obrazu całego obiektu (np. samolotu) w rzucie prostokątnym, a następnie stworzenie możliwości uzyskania rzutu perspektywicznego, niezbędnego do animacji obrazu. Przy analizie metody i tworzeniu programu przyjęto założenie, że obiekty będą opisywane za pomocą przekrojów i ścian, następnie jednak zastosowano również trzeci sposób opisu obiektów, za pomocą wielokątów. Istotnym czynnikiem warunkującym przyjęcie struktury danych była konieczność zapewnienia możliwości definiowania obiektów dowolnie skomplikowanych, występujących w połączeniu z innymi obiektami. Konsekwencją wymienionych założeń było przyjęcie struktury hierarchicznej, pozwalającej na tworzenie obiektów dowolnie skomplikowanych za pomocą obiektów prostszych. Dzięki temu można tworzyć poszczególne fragmenty, po czym łączyć je, i w ten sposób tworzyć złożony obiekt. Opracowano i przebadano szereg procedur i funkcji pozwalających spełnić postawione założenia. Do reprezentacji struktury w pamięci zastosowano zmienne dynamiczne. Są nimi listy, którymi można dość elastycznie manipulować. Z listami nie wiążą się ograniczenia co do ilości elementów, z jakich muszą się składać. Tablica w momencie deklarowania wymaga określenia jej rozmiaru, natomiast w momencie definiowania, ten wymiar nie jest znany. Lista stanowi najbardziej przystępną strukturę, chociaż powoduje większą zajętość pamięci, gdyż niezbędne są dodatkowe pola będące wskazaniem na kolejne elementy listy [5].

Cały program testujący algorytm zapisu i przyjętą metodę został napisany przy użyciu struktury modułowej. Miało to umożliwić zmniejszenie wielkości pojedynczego pliku w celu zapewnienia jego lepszej czytelności. Obiekty opisywane są w sposób hierarchiczny, tzn. definiowanie następuje od poziomu najniższego, gdzie stosowany jest opis za pomocą najprostszyc elementów takich jak: przekrój, ściana, wielokąt. Elementy te opisują obiekty takie jak np.: skrzydło, kadłub. Następnie następuje przejście na wyższy poziom, w którym opisane są bardziej złożone obiekty. Powstają one przez połączenie obiektów niższego poziomu. Tak można "skleić" np. statecznik poziomy ze statecznikiem pionowym, skrzydła z kadłubem itp. Kolejne wyższe poziomy mogą opisywać bardziej złożone obiekty typu samolot, bądź kilka samolotów, lub całą scenę obrazującą sytuację na lotnisku. Ten sposób definiowania struktury daje możliwość tworzenia scen, takich jak lotnisko z pojazdami poruszającymi się po pasie startowym i samolotami w powietrzu. Samolot (statek powietrzny) może być zdefiniowany jako element niższego poziomu, natomiast na wyższym poziomie, jest on zdefiniowany za pomocą czterech zmiennych. Są nimi: nazwa samolotu, położenie w przestrzeni, rotacja opisująca jego zorientowanie wokół osi oraz skala.

Komunikacyjne statki powietrzne pojawiające się w obszarze kontrolowanym FIR Warszawa charakteryzują się odpowiednimi parametrami techniczno-ruchowymi. W celu wizualizacji sytuacji ruchowej konieczne było dokonanie analizy porównawczej tych charakterystyk w celu opracowania standardowych typów statków powietrznych (samolotów). Typy te zostały określone podczas analizy ruchu lotniczego. Ich wybrane parametry techniczno-ruchowe poddano dekompozycji i podzielono na kategorie. Dla poszczególnych kategorii opracowano charakterystyki standardowe. Posługując się charakterystykami standardowymi można zaplanować:

- wzorcowe trajektorie lotu,
- manewry zmiany poziomów lotu,

- czasy wykonania manewrów,
- optymalne prędkości lotu i ich zmiany dla zadanych separacji.

### 5. Model opisu danych układu wizualizacji

Dane opisujące obiekty lub sceny na lotnisku mają strukturę hierarchiczną, tzn. obiekt, wyższego poziomu są opisywane za pomocą elementów podstawowych lub obiektów niższego poziomu. Możemy wyróżnić zbiór elementów podstawowych  $E_p$  składający się z trzech podzbiorów:

$$E_p = \Pi \cup \Omega \cup \Sigma$$

gdzie:

$\Pi$  - zbiór elementów  $\pi_i$  typu przekrój,

$\Omega$  - zbiór elementów  $\omega_i$  typu wielokąt,

$\Sigma$  - zbiór elementów  $\sigma_i$  typu ściana.

Można także określić zbiór elementów opisowych  $E_o$  składający się z dwóch podzbiorów:

$$E_o = B \cup M$$

gdzie:

B - zbiór elementów  $\beta_i$  typu BigBOX,

M - zbiór elementów  $\mu_i$  typu SmallBOX.

Na zbiorach elementów podstawowych  $E_p$  możemy określić operację opis przekształcającą go na zbiór elementów opisowych M:

$$\Pi \xrightarrow{\text{opis}} M$$

$$\Omega \xrightarrow{\text{opis}} M$$

$$\Sigma \xrightarrow{\text{opis}} M$$

Na zbiorach elementów opisowych  $E_o$  można natomiast określić operację powiązania, która przekształca go na zbiór elementów opisowych B:

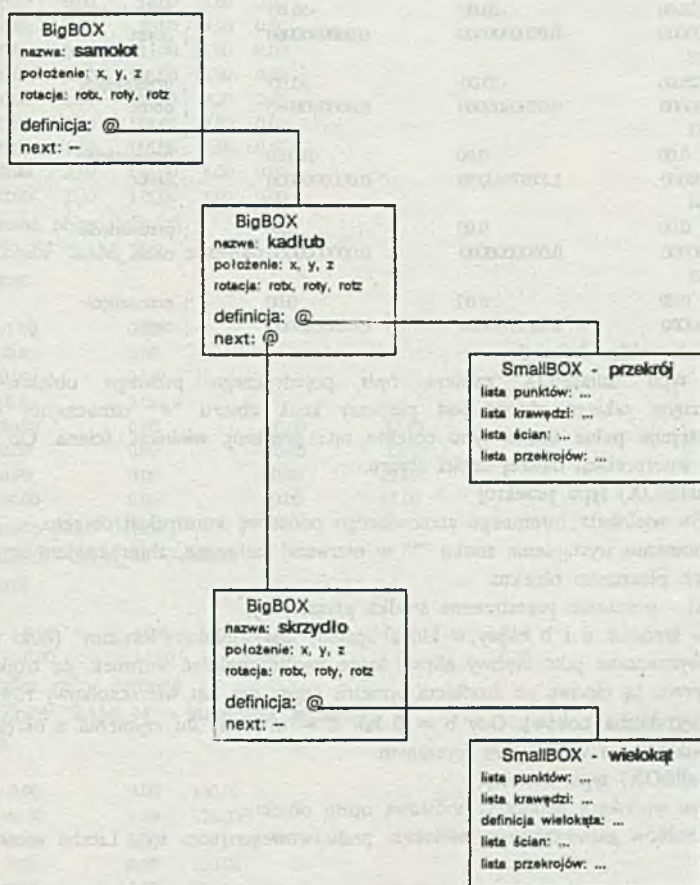
$$B \xrightarrow{\text{powiazanie}} B$$

$$M \xrightarrow{\text{powiazanie}} B$$

W efekcie wykonania tych operacji otrzymujemy jeden element złożony typu opisowego B, który definiuje całą scenę. Przykładowy schemat opisu obiektu złożonego "samolot" przedstawia rys.1. Można postawić problem znalezienia takiego sposobu opisu sceny graficznej, aby zminimalizować złożoność opisu sceny przy jednoczesnym zachowaniu założonego stopnia podobieństwa do opisywanego pierwowzoru. Problem ten sprowadza się w istocie do wyboru odpowiednich elementów podstawowych  $\pi, \omega, \sigma$  oraz opisowych  $\beta$  i  $\mu$ , a także do określenia, które z tych elementów poddać działaniu operatorów opis i powiązanie. Nie jest możliwe w chwili obecnej znalezienie rozwiązania optymalnego takiego zadania, ze względu na jego znaczną złożoność oraz na trudność w jego formalnym postawieniu. Prezentowany w niniejszym artykule sposób opisu sceny graficznej jest próbą znalezienia suboptymalnego rozwiązania tego zagadnienia.

Schemat przedstawiony na rys.1 obrazuje strukturę opisu dwóch obiektów niższego poziomu, jakimi są kadłub samolotu i skrzydło. Poprzez dodanie tylko jednego elementu wyższego poziomu, znajdującego się nad opisem elementów niższego poziomu, otrzymujemy opis obiektu złożonego. Kadłub opisany jest kostką SmallBOX, będącą kostką niższego poziomu, a dokonującą opisu obiektu dzięki liście punktów, liście krawędzi, liście ścian oraz liście przekrojów, niezbędnych do zdefiniowania obiektu typu przekrój. Kostka wyższego poziomu BigBOX podaje nazwę obiektu, określa położenie obiektu w przestrzeni oraz wskazuje kolejną

kostkę wyższego poziomu BigBOX, opisującą skrzydło. Skrzydło opisane jest kostką SmallBox, będącą kostką niższego poziomu, a dokonującą opisu dzięki liście punktów, liście krawędzi i liście ścian, niezbędnych do zdefiniowania obiektu typu przekrój. Kostka wyższego poziomu podaje nazwę skrzydło oraz określa położenie obiektu w przestrzeni. Oba obiekty są więc już opisane. Można uznać je za obiekty niższego poziomu dla obiektu, który miały być złożony z ich obu. Do opisanie obiektu samolot (kadłub + skrzydło) potrzebna jest jeszcze jedna kostka wyższego poziomu BigBOX, pozwalająca na nadanie nazwy takiemu obiektowi oraz na określenie położenia tego obiektu w przestrzeni. Kostka wyższego poziomu BigBOX może łączyć dowolną ilość elementów niższego poziomu. Może też istnieć dowolna ilość wyższych poziomów. Jedynym ograniczeniem pozostaje zapewnienie wystarczającej pamięci przechowującej opis skomplikowanych scen.



Rys1. Przykładowy schemat opisu obiektu złożonego "samolot".

Fig.1. Schematic diagram of describing complex element "airplane".

## 6. Przykłady danych opisujących elementy sytuacji na lotnisku

Opis zbioru typu BigBOX. Znak "\$" w pierwszej kolumnie oznacza, że w danym wierszu umieszczona jest nazwa zbioru zawierające dodatkowe informacje o obiekcie, takie jak następny zbiór typu BigBOX lub zbiór typu SmallBOX, opisujący konkretny element. Następujące dalej dwa wiersze opisują:

- 1) przesunięcie wzdłuż osi (x, y, z) całego obiektu opisanego w zbiorze pochodnym,
- 2) obrót wokół osi (x, y, z) całego obiektu pochodnego.

Zbiór "SAM666" typu BigBOX opisujący cały obiekt.

SSAM_33				
0.00	0.00	0.00		; przesunięcie
0.0000000000	0.0000000000	0.0000000000		; obrót
SSAM_33				
28.00	-20.00	-55.00		; przesunięcie
0.0000000000	0.0000000000	0.0000000000		; obrót
SSAM_33				
28.00	-20.00	-55.00		; przesunięcie
0.0000000000	0.0000000000	0.0000000000		; obrót
SSAM_22				
0.00	0.00	100.00		; przesunięcie
0.0000000000	1.5707963270	0.0000000000		; obrót
SSAM_44				
0.00	0.00	0.00		; przesunięcie
0.0000000000	0.0000000000	0.0000000000		; obrót
SSAM_55				
0.00	0.00	0.00		; przesunięcie
0.0000000000	0.0000000000	0.0000000000		; obrót

\*\*\* koniec pliku definicji

Zbiór typu SmallBOX zawiera opis pojedynczego prostego obiektu. Elementem charakterystycznym takiego zbioru jest pierwszy znak zbioru "#" oznaczający opis obiektu, następnie występuje pełna nazwa typu obiektu, np.: przekrój, wielokąt, ściana. Od typu obiektu zależy sposób interpretacji dalszej części zbioru.

OBIEKT (SmallBOX) typu przekrój -

- 1) liczba boków wielokąta foremnego stanowiącego podstawę konstrukcji obiektu,
- 2) dalej do momentu wystąpienia znaku "\*" w pierwszej kolumnie, zbiór zawiera opis położenia kolejnych płaszczyzn obiektu:
  - (x, y, z) - położenie przestrzenne środka płaszczyzny,
  - (a, b) - średnice a i b elipsy, w którą wpisany jest wielokąt "foremny" (boki wielokąta są wyznaczone jako cięciwy elipsy, które muszą spełniać warunek, że trójkąt utworzony przez tę cięciwę ze środkiem symetrii elipsy ma kąt wierzchołkowy równy  $360^\circ/\text{liczba boków}$ ). Gdy  $b = 0$  lub  $b = a$ , mamy do czynienia z okręgiem i klasycznym wielokątem foremnym.

OBIEKT (SmallBOX) typu wielokąt -

- 1) liczba boków wielokąta będącego podstawą opisu obiektu,
- 2) opis wierzchołków jednostkowego wielokąta podstawowego (wsp. x,y). Liczba wierszy określona w p.1.
- 3) dalej do momentu wystąpienia znaku "\*" w pierwszej kolumnie, zbiór zawiera opis położenia kolejnych płaszczyzn obiektu:
  - (x, y, z) - położenie przestrzenne środka płaszczyzny,
  - a - wartość określająca wielkość wielokąta podstawowego.

OBIEKT (SmallBOX) typu ściana -

- 1) liczba punktów opisujących obiekt,



- 2) opis kolejnych punktów (wsp. x, y, z). Liczba wierszy określona w p.1,  
 3) dalej do momentu wystąpienia znaku "\*\*\*" w pierwszej kolumnie, zbiór zawiera opis kolejnych płaszczyzn. Płaszczyzna definiowana jest przez 4 punkty. Wiersz opisujący płaszczyznę zawiera numery definiujących ją punktów.

Dalej podano przykłady zbiorów opisujących obiekty.

Zbiór "SAM\_11" - kadłub

#przekrój

12

0.00	2.50	0.00	1.00	0.00
0.00	1.50	3.00	3.50	0.00
0.00	1.50	5.00	4.50	0.00
0.00	1.00	6.00	5.50	0.00
0.00	0.50	8.00	6.25	0.00
0.00	0.50	10.00	7.00	0.00
0.00	0.00	14.00	8.00	0.00
0.00	0.00	28.00	9.50	0.00
0.00	0.00	35.00	10.00	0.00
0.00	0.00	126.00	10.00	0.00
0.00	1.00	135.00	9.20	0.00
0.00	1.50	143.00	8.50	0.00
0.00	1.50	155.00	6.50	0.00
0.00	2.50	166.00	5.00	0.00
0.00	3.00	175.00	1.00	0.00

\*\*\* koniec pliku definicji

Zbiór "SAM\_22" - skrzydło

#wielokąt

4

-1.00	0.00		
-0.83	0.08		
0.00	0.00		
-0.83	0.08		
30.00	0.00	80.00	15.00
0.00	0.00	20.00	25.00
0.00	0.00	-20.00	25.00
30.00	0.00	-80.00	15.00

\*\*\* koniec pliku definicji

Zbiór "SAM\_33" - silnik

#przekrój

8

0.00	16.00	135.00	4.50	0.00
0.00	16.00	170.00	4.50	0.00

\*\*\* koniec pliku definicji

Zbiór "SAM\_44" - statecznik poziomy

#ściana

12

0.00	0.00	160.00
-40.00	6.00	170.00
-40.00	6.00	155.00
0.00	0.00	130.00
40.00	6.00	155.00
40.00	6.00	170.00
0.00	2.00	130.00
-40.00	7.00	155.00
-40.00	7.00	170.00
0.00	2.00	160.00

40.00	7.00	155.00	
40.00	7.00	170.00	
0	1	2	3
4	5	0	3
6	7	8	9
10	6	9	11

\*\*\* koniec pliku definicji

Zbiór "SAM\_55" - statecznik pionowy

#ściana

8

-1.00	0.00	125.00	
-1.00	50.00	165.00	
-1.00	50.00	185.00	
-1.00	0.00	165.00	
1.00	0.00	125.00	
1.00	50.00	165.00	
1.00	50.00	185.00	
1.00	0.00	165.00	
0	1	2	3
4	5	6	7

\*\*\* koniec pliku definicji

## 7. Implementacja programowa metody zobrazowania elementów rejonu lotniska dla symulatora TWR

Opracowany program testujący metodę pracuje w dwóch trybach: podstawowym - budowy sceny złożonej (1), oraz trybie definicji obiektu prostego (2).

W trybie podstawowym scena złożona jest budowana przez dodawanie kolejnych obiektów podstawowych lub złożonych do definicji edytowanej sceny. Podaje się nazwę zbioru, z którego należy pobrać definicje obiektu oraz współrzędne  $(x,y,z)$  obiektu na scenie. Możliwe jest przejście do trybu definicji obiektu prostego. Przejście podczas edycji sceny w tryb (2) powoduje utratę obecnie edytowanej sceny. Można także dokonać:

- przesunięcia pojedynczego elementu względem pozostałych elementów poprzez wybranie elementu i podanie nowych współrzędnych  $(x,y,z)$ ,
- obrotu pojedynczego elementu względem pozostałych elementów poprzez wybranie elementu i podanie nowych kątów obrotu wokół osi  $x, y, z$ :  $(v_x, v_y, v_z)$ .

W trybie definicji obiektu prostego możliwe jest wprowadzanie obiektów typu: przekrój, wielokąt, ściana. Elementy dodaje się do zbioru definicji na trzy sposoby:

- gdy element jest typu przekrój - podaje się położenie przestrzenne środka dodawanej płaszczyzny oraz średnice  $a$  i  $b$  elipsy, w którą wpisany jest wielokąt;
- gdy element jest typu wielokąt - podaje się położenie przestrzenne środka płaszczyzny oraz rozmiar kwadratu, w który wpisany jest wielokąt podstawowy;
- gdy element jest typu ściana - podaje się położenie czterech punktów definiujących ścianę.

W obu trybach pracy dostępne jest ponadto skalowanie obiektu oraz zmiana punktu położenia obserwatora.

## 8. Podsumowanie

Podstawowym problemem powstałym w trakcie prowadzenia prac badawczych, było ograniczenie szybkościowe stosowanych komputerów i ich kart graficznych. Prace badawcze były prowadzone na dwóch zestawach komputerowych typu IBM PC:

zestaw 1 - IBM AT/386 z kartą graficzną SVGA

zestaw 2 - IBM AT/386 z kartą GEMINI 118.

W przypadku zestawu 1 "wąskim gardłem" był proces rysowania, zaś dla zestawu 2 - proces obliczania współrzędnych. W wyniku realizacji przedstawianego zadania badawczego można

sprecyzować następujące wnioski dotyczące metod wizualizacji wybranych elementów ruchu lotniczego dla symulatora TWR:

- konieczne jest zastosowanie zestawów komputerowych o maksymalnie dużej mocy obliczeniowej, np: 486/50MHz, o pełnej 32bitowej architekturze oraz pamięci rzędu 8-32MB;
- wskazane jest zastosowanie kart graficznych typu GEMINI z wewnętrznym procesorem graficznym i o dużej pamięci wewnętrznej rzędu 2-3MB lub kart graficznych wyposażonych w dodatkowy procesor zarządzający;
- należy kontynuować prace nad optymalizacją algorytmów przetwarzania danych obrazowych;
- konieczne jest wykorzystanie do pisania programów graficznych systemu operacyjnego dla komputera w pełni 32bitowego (np: OS/2 2.0, XENIX, UNIX, VMS, VRTX itp.), wykorzystującego pełne możliwości procesora oraz kompilatorów języków wysokiego poziomu jak Pascal, Ada, Modula-2.

Sugestia dotycząca wykorzystania zestawu wyposażonego w procesor I80486 wiąże się z popularnością tego typu sprzętu na polskim rynku, tym niemniej możliwe i w pełni uzasadnione byłoby zastosowanie komputerów zbudowanych w oparciu o inny typ procesora, np: Motorola 68040 czy VAX serii 6000. Napisanie całości oprogramowania w językach wysokiego poziomu i w dodatku bez wykorzystywania specyficznych cech danej implementacji kompilatora języka wysokiego poziomu pozwoliłoby na łatwą zmianę zarówno systemu operacyjnego, jak też i sprzętu, w oparciu o który będzie wykonany symulator.

Temat, którego wstępną analizę przedstawiono w tym artykule, będzie kontynuowany w następnych latach w celu doprowadzenia do uruchomienia szkoleniowo-badawczego symulatora TWR.

## LITERATURA

- [1] Angell I.: Wprowadzenie do grafiki komputerowej. WNT, Warszawa, 1988,
- [2] Atack M.J. A simulation model for calculating annual congestion delay arising from airport runway operations. Journal of the Operational Research Society, Vol.29. No 4, April 1978.
- [3] Franklin W. R.: A linear exact hidden surface algorithm. SIGGRAPH'80, Seattle, Washington, July 1980, s.117-123.
- [4] Friedman F. On the properties of the sum of durations of perceived conflicts in air traffic control. Math. and Comput. Simul. Vol.28, No 4, 1986.
- [5] Herman G.T., ed.: Image reconstruction from projections. Berlin, Springer 1980.
- [6] Jankowski M. :Elementy grafiki komputerowej, WNT, Warszawa 1990,
- [7] Knowlton K. : Computer-aided definition, manipulation and description of objects composed of spheres. ACM Computer Graphics, 1981, No.15 s.48-71.
- [8] Lane J. M., Riesefeld R.F.: A theoretical development for the computer generation and display of piecewise polynomial surfaces. IEEE Trans. on Pattern Analysis and Machine Intelligence, No.2, 1980, s.35-46.
- [9] Pavlidis T.: Grafika i przetwarzanie obrazów, WNT, Warszawa 1987,
- [10] Podlasiak P.: Metody komputerowej generacji obrazów ruchomych w zastosowaniu do symulatorów lotu. PW, Warszawa 1984,
- [11] Szczepanski C.: Symulatory lotu, WPW, Warszawa 1990,
- [12] -: Flight Simulator Design and Performance Data Requirements, IATA, 1986.

Recenzent: Prof.dr h.inż. Eugeniusz Toczyłowski  
Wpłynęło do Redakcji do 30.04.1992 r.

### Abstract:

In this paper a method of describing complex elements (such as airplane) used for the computer animation is presented. A computer program for data input (describing parts of airplanes and airport facilities

according to a given structure) is elaborated. Data is processed to obtain orthogonal projection of all parts along x,y,z axis. This simplifies edition while defining the object. Then all parts are joined to get a view of whole object (for instance aircraft) in orthogonal projection. A possibility to get perspective projection that is necessary for computer animation is worked out. Hierarchical structure of data is assumed. It makes possible to construct more complicated objects using simple ones. Theoretical basis of such hierarchical objects transformations are analysed.

The main problem that has been found during research work are constrains connected with computer speed.