

ПАКЕТ ПРИКЛАДНИХ ПРОГРАМ ВІЗУАЛІЗАЦІЇ ПОВІТРЯНОЇ ОБСТАНОВКИ В НАВІГАЦІЙНИХ ГЕОІНФОРМАЦІЙНИХ КОМПЛЕКСАХ РЕАЛЬНОГО ЧАСУ

М.І.Васюхін, В.Д.Гулевець, А.М.Касім, О.Л.Бойко
Національний авіаційний університет

У статті описується пакет прикладних програм візуалізації повітряної обстановки в навігаційних геоінформаційних комплексах реального часу, а також наводяться основні алгоритми, які покладено в основу розробленого пакету програм.

Вступ. Сфера застосувань навігаційних геоінформаційних комплексів реального часу (НГК РЧ) достатньо широка – від систем, що забезпечують безпеку польотів, тренажерів різних транспортних засобів (літаки, космічні кораблі, гелікоптери й ін.) до потужних центрів оперативного керування всіх рівнів і призначення, які використовуються для відображення на карті швидкоплинних процесів, пов'язаних з візуалізацією повітряної та наземної обстановки у вигляді динамічних сцен.

Після катастрофи в Чорнобилі надзвичайна важливість і необхідність автоматизованих систем екологічного моніторингу, що містять у своєму складі геоінформаційну систему (ГІС), стала очевидною. Значну увагу створенню подібних систем протягом останніх 20 років приділяють практично всі високорозвинені країни.

ГІС для НГК РЧ є джерелом, що забезпечує надання картографічної інформації, необхідної для побудови динамічних сцен, які відображають на її фоні еволюції об'єктів, що рухаються в навколосемному просторі, у реальному часі. Побудова таких сцен дає можливість фахівцям адекватно сприймати навколишнє оточення й оперативно приймати відповідні рішення.

Типовим прикладом застосування НГК РЧ в авіації є комплекс, що містить систему диспетчерської служби та призначений для відображення положення літаків, що перебувають навколо великого аеропорту, й для підтримки прийняття оперативних рішень при розв'язанні задачі безпечного розведення літаків. Збільшення останнім часом випадків авіакатастроф говорить про нагальну потребу широкого впровадження таких комплексів для забезпечення безпеки польотів.

Завдання керування військами, запобігання терактів і т. п., що розв'язуються, наприклад, у сучасних системах протиповітряної й протиракетної оборони (ППО й ПРО), потребують використання НГК РЧ. У цьому випадку вирішується задача, обернена безпечному розведенню літальних апаратів – так звана задача переслідування. З огляду на значні швидкості літаків і ракет задача переслідування при її поданні й вирішенні в реальному часі значно складніша за задачі розведення.

Основні вимоги, що висуваються при створенні навігаційного геоінформаційного комплексу з функціонуванням в режимі реального часу – досягнення максимальної швидкодії при забезпеченні «динаміки» символів на картографічному фоні – обумовлені великими обсягами потоків картографічних даних (сотні мегабайт) і рівнем надійності, необхідним для прийняття адекватних рішень людиною-оператором. Відомо, що для побудови систем вказаного класу використовуються різні підходи, спільний недолік яких – відсутність раціонального узгодження теоретичних моделей і програмно-апаратних засобів для їх реалізації.

Постановка проблеми. Не можна не відзначити ще один момент. Незважаючи на деяке потепління міжнародної обстановки, військовий фактор продовжує відігравати важливу роль у світовій політиці. Всі великі держави світу продовжують удосконалювати свої збройні сили. Насамперед це стосується новітнього озброєння й військової техніки, широкого застосування інформаційних технологій в цій галузі. Іде постійний активний пошук нових засобів збройної боротьби.

Безпека країни – не тільки в потужних Збройних Силах, але й, насамперед, у досконалості програмно-технічних засобів центрів оперативної взаємодії, з'єднаних інформаційними каналами. Сьогодні не можна не відзначити зростаючу залежність нашої країни від закордонних партнерів у засобах інформатики й зв'язку. І от чому. Маначинський [1] так представляє ймовірний американський варіант сценарію війни ХХІ століття. У випадку, якщо сучасний Вавилон (можливо, це Кабул, Багдад або Тегеран) раптово нападає на США або одного з їхніх союзників, то США, з одночасним перекиданням своїх військ і кораблів у бойовий район дій для захисту свого стратегічного союзника, починають інформаційну війну проти агресора. На самому початку конфлікту оживають комп'ютерні віруси й логічні бомби, заздалегідь таємно закладені у пам'ять ЕОМ, застосовуваних у всіх структурах державного, військового й економічного управління країни-агресора. Ця зброя пускається в хід

спеціальною командою, наприклад, із супутника або через INTERNET. Після початку конфлікту агенти спецслужб за допомогою портативних, потужних генераторів руйнують програмне забезпечення в системах цивільного й військового керування. Одночасно із цим по міжнародних телекомунікаційних мережах обнуляються банківські рахунки агресора. Відразу придушуються теле- і радіопередачі всіх станцій на території країни-нападника, всіх пунктів зв'язку й керування військами агресора та починають передаватися матеріали, спрямовані на дестабілізацію політичної обстановки, дезорієнтацію населення й поширення паніки. Ці дії тягнуть для агресора катастрофічні наслідки, оскільки повністю дезорганізують роботу всіх життєво важливих для країни систем. Найбільш важливими об'єктами є, в першу чергу, система керування державою, командні пункти ПРО й ППО, тощо.

У результаті такої комбінованої атаки за допомогою різних видів інформаційної зброї в країні-агресорі панує хаос, відбувається колапс економіки, різко погіршується соціально-політична ситуація. Ефективність застосування такої зброї підтверджують операції «Буря в пустелі» і, особливо, «Лис у пустелі». Агресор змушений або відмовитися від своїх планів або підкоритися вимогам США та їхніх союзників.

У сучасних умовах інформаційна зброя стає найважливішою частиною військового й військово-пропагандистського потенціалів США та їхніх союзників. Якщо в 1980 році на розвиток інформаційних технологій ними витрачалось біля мільярда доларів, то в 1994 році – уже понад 25 млрд. дол., а в подальшому ця цифра ще зростала. Придбання інформаційних технологій за обсягами фінансування вийшло на перше місце серед програм озброєнь США, помітно випереджаючи навіть космічні й ракетно-ядерні програми, й в 2000 р. перевищило 100 млрд. дол.

Становлення України як незалежної держави, прагнення якнайшвидше залучитися до світового рівня інформаційних технологій призвели до нібито виправданої ставки на використання телекомунікаційних мереж, що існують у розвинених країнах. При цьому західні інформаційні мережі зі своїми стандартами передачі даних і апаратно-програмних засобів були уведені в межі України. Вже можна стверджувати, що іноземні телекомунікаційні мережі фактично зайняли пануюче положення на території України, стерши міждержавні кордони в інформаційному просторі.

Масовий перехід на імпорتنу техніку й закордонні системи є небезпечним для України, оскільки виходить, що система – у нас, а

ключ до неї – у них. Здобуваючи й впроваджуючи за чималі гроші готові інформаційні технології й апаратно-програмні засоби в надії стати врівень із розвиненими країнами, Україна стає заручником цієї політики, бо тепер керування процесами, що відбуваються в інформаційному просторі, майже не залежить від структур керування державою, тому що, впливаючи на інформаційні потоки, можна управляти всією країною.

У зв'язку з цим економічну й науково-технічну політику держави варто розглядати через призму інформаційної безпеки. Будучи відкритою, орієнтованою на дотримання законних прав громадян на інформацію й інтелектуальну власність, ця політика не може не бути протекціоністською. Вона повинна підтримувати вітчизняних виробників інформаційних технологій, захищати внутрішній ринок від проникнення в нього схованих елементів інформаційної зброї. Це особливо важливо сьогодні, коли здійснюються масові закупівлі інформаційних технологій за кордоном, а власний науково-технічний і виробничий потенціал поступово руйнується.

Основна частина. Як зазначалося вище, до характерних задач реального часу, властивим НГК РЧ, відносяться задача безпеки польотів – розведення літаків в районі крупних аеропортів, та зворотна їй – задача переслідування.

Задача переслідування - утікання є однією з основних та найважчих задач, притаманних даним комплексам. Початок теоретичним дослідженням таких задач був встановлений Р. Айзексом [2] та продовжений Л.С. Понтрягіним [3], Н.Н. Красовським і В.Д. Батухтіним [4, 5], а також Б.Н. Пшеничним [6] та його учнями.

В основу вирішення практичної задачі переслідування покладено метод дозволяючих функцій, розвинутий А.А. Чикрієм [7, 8], який дає обґрунтування класичного правила паралельного переслідування. Основною причиною такого вибору є практична спрямованість методу та можливість чисельної реалізації процесу переслідування.

Одна з програм запропонованого пакету прикладних програм (ППП) реалізує задачу переслідування ракетою літака, який рухається по задалегідь заданому маршруту на картографічному фоні, рис.1.

Опишемо постановку задачі переслідування в загальному вигляді. Нехай рух керованого об'єкту в просторі R^n описується рівнянням

$$\dot{z} = Az + \varphi(u, v), \quad (1)$$

де z - фазовий вектор, який складається з геометричних координат, швидкостей, прискорень переслідувача та втікача; A - квадратна матриця порядку n , $\varphi(u, v), \varphi: U \times V \rightarrow R^n$ - неперервна за сукупністю змінних функція - блок управління; u і v - параметри управління переслідувача й втікача, що вибираються з областей управління U і V , які є непорожніми компактами простору R^n (замкнута і обмежена множина в R^n - компакт). R^n - евклідовий n -вимірний простір точок $x = (x_1, x_2, \dots, x_n)$.

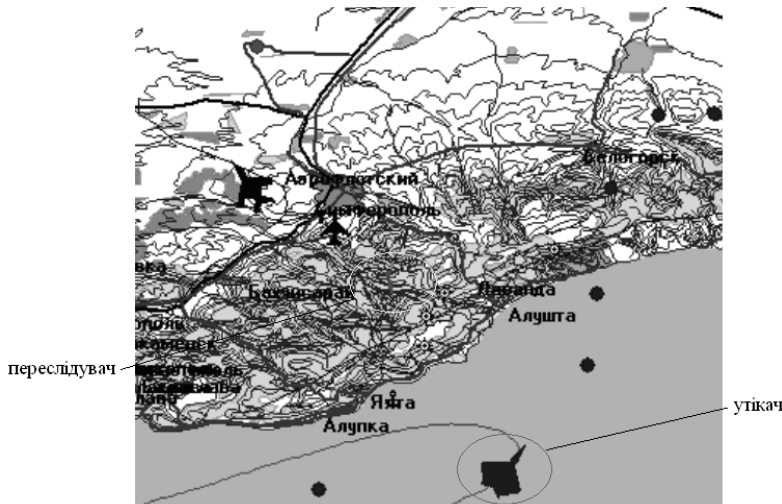


Рис. 1. Кадр сцени переслідування ракетою літака на картографічному фоні.

Вибираючи свої управління у вигляді деяких функцій, кожен з гравців впливає на процес (1), переслідуючи свої цілі. Мета переслідувача - вивести траєкторію процесу (1) на деяку замкнуту множину M^* , яку А.А. Чикрій назвав термінальною, за найкоротший час, а мета втікача - відхилити траєкторію процесу (1) від зустрічі з множиною M^* на всьому напівнескінченному інтервалі часу або, якщо це неможливо, максимально відтягнути момент зустрічі. Працездатність методу А.А. Чикрія [7] ілюструватиметься на простому прикладі, коли об'єкти управляються за швидкістю.

Тут рух переслідувача та втікача описується рівняннями

$$\begin{aligned} \dot{x} &= u, \quad \|u\| \leq a, \quad a > 1, \quad x \in R^n, \\ \dot{y} &= v, \quad \|v\| \leq 1, \quad y \in R^n. \end{aligned} \quad (2)$$

Цим рівнянням руху відповідають рухи з фіксованою швидкістю.

Термінальна множина задається рівністю $x = y$, тобто переслідуювач наздогнав об'єкт переслідування. Зробивши заміну $z = x - y$, процес (2) зводиться до вигляду

$$\dot{z} = u - v, \quad z \in R^n, \quad \|u\| \leq a, \quad \|v\| \leq 1. \quad (3)$$

У випадку простого руху дозволяюча функція для $z \neq 0$ задається виразом

$$\alpha(z, v) = \frac{(z, v) + ((z, v)^2 + \|z\|^2 (a^2 - \|v\|^2))^{1/2}}{\|z\|^2}. \quad (4)$$

А управління переслідуювача має вигляд

$$u(\tau) = v(\tau) - \alpha(z^0, v(\tau))z^0, \quad \tau \in \left[0, \frac{\|z^0\|}{a-1}\right], \quad (5)$$

де z^0 – початковий стан процесу.

При законі управління переслідуювача (4), (5) для траєкторії процесу (3) реалізується паралельне переслідування. Інакше кажучи, $z(t)$ паралельно z^0 для всіх t , таких, що

$$1 - \int_0^t \alpha(z^0, v(\tau)) d\tau > 0.$$

Отже, стратегія, яка приписує переслідуювачу закон управління (4), (5), є стратегією паралельного переслідування, яка полягає в тому, що в процесі переслідування прямі, які сполучають поточні положення переслідуювача та втікача (лінії візування, рис. 2), повинні бути паралельними.

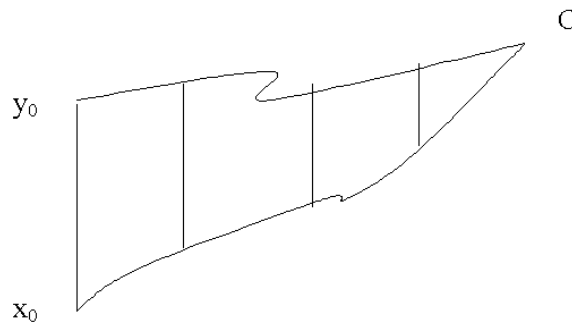


Рис. 2. Траєкторії польоту цілі та переслідуювача.

На рис. 2 зображено траєкторії польоту цілі та переслідуювача. Переслідуювач наздоганяє втікача в точці O . Крім того, там показано зміни параметра z за періодичні інтервали часу.

Для реалізації методу, що дозволяє побудувати стратегію, яка б задавала переслідуювачу закон управління (4), (5), пропонується наступний алгоритм [9], рис. 3.

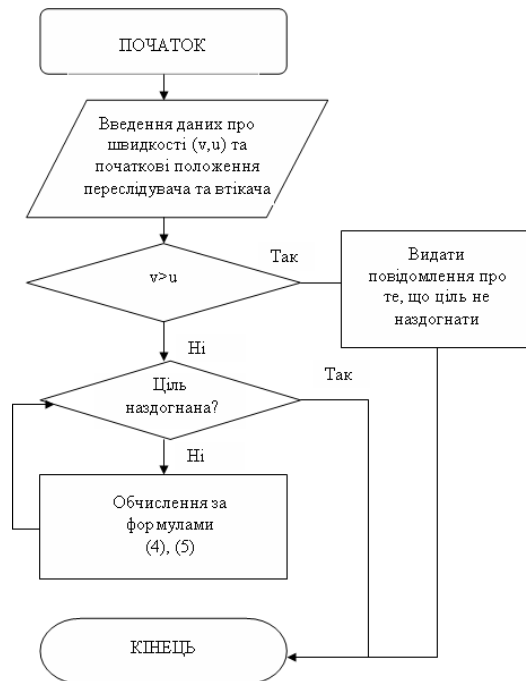


Рис. 3. Алгоритм переслідування цілі.

В цьому алгоритмі час розбивається на деякі проміжки тривалістю t і, враховуючи те, що рівняння руху подані в диференціальній формі, то їх можна застосувати на кожному з цих проміжків. Ці формули дозволяють обчислити необхідну швидкість управління переслідувача. Звідси, інтегруючи, ми одержуємо координати переслідувача і втікача. Запропонований алгоритм повторюється, поки переслідувач не досягне цілі, рис. 2, точка O .

Програма для випадку переслідування буде наступною:

```

program Presledovanie;
var xv,xv,yu,yv,u,v,u1,u2,v1,v2 : real;
var xu1,xv1,yu1,yv1,mz,a,x,t,tk : real;
var k,sw : integer;
begin
  readln(xv);readln(yv);readln(v); {введення даних}
  readln(xu);readln(yu);readln(u);
  t:=0.01;sw:=1;
  xv1:=xv;xu1:=xu;yu1:=yu;yv1:=yv;      {встановлення початкових
  параметрів циклу}
  if xv<xu then
    begin xv:=-xv;xu:=-xu;sw:=-1;
    end;
  
```

```

if v>u then writeln('ne dognat'); {перевірка умови можливості
досягнення цілі}
if v<u then
begin
k:=0;
while xv1>xu1 do begin
v1:=v*3/5;v2:=v*4/5; {вводиться будь-яка функція руху, яка
задовольняє умовам на норму (2)}
x:=(xu-xv)*v1+(yu-yv)*v2; {обчислення за формулами}
mz:=(xu-xv)*(xu-xv)+(yu-yv)*(yu-yv);
a:=(x+sqrt(x*x+mz*(u*u-v*v)))/(mz);
u1:=v1-a*(xu-xv);
u2:=v2-a*(yu-yv);
xv1:=xv1+v1*t;yv1:=yv1+v2*t;
xu1:=xu1+u1*t;yu1:=yu1+u2*t;
writeln(xu1,xv1,yu1,yv1,k);k:=k+1;
end;
end;
tk:=t*k;
writeln(sw*xu1,yu1,tk);
end.

```

Для значень параметрів польоту літака, наприклад: висота 20 км, швидкість 300 м/с (1080 км/год), старту ракети-перехоплювача на відстані 150 км та її швидкості 1км/с за умови маневру цілі час піймання за даним алгоритмом складе 2 хв. Розрахунковий час співпадає з реальним часом піймання, що дозволяє рекомендувати цей метод для практичного використання в НГК РЧ.

Загальною метою розробки ППП, що забезпечує побудову динамічних сцен, була реалізація методів й алгоритмів для обробки двовимірних зображень символів динамічних об'єктів, що рухаються в навколоремному просторі, з дотриманням режиму реального часу. В даному випадку під обробкою зображень розуміється організація плавного переміщення, швидкого повороту символів, наприклад, літаків, що описуються матрицями пікселів різної розмірності, від 8x8 до 128x128, – у випадку подання символів в растровому вигляді і що будуються з графічних примітивів (ліній та еліпсів) за алгоритмами несиметричного цифрового диференціального аналізатора (ЦДА) і Брезенхема – у випадку їх представлення у векторному вигляді [10].

При написанні другої програми розробленого ППП, яка моделює поступально-обертальне переміщення символу літака із

забезпеченням скролінгу картографічного фону, нами використовувався підхід, суть якого полягає в наступному.

Є графічна база даних (БД) символів літаків різної азимутальної орієнтації, зображення яких представлені в растровому вигляді матрицями розміром 32x32. Всього в БД зберігається 16 символів, повернених на градусну величину, кратну 22,5°. Вважається, що картографічний фон вже сформований заздалегідь. Після визначення перших координат об'єкту-літака, представлено складним відеосимволом для відображення його в заданій точці на карті з БД символів «завантажується» символ в певному напрямку, який вказує курс польоту літака. Управління рухом символу здійснюється за допомогою клавіатури: клавішею "W" запускається рух символу, клавішею "S" рух символу зупиняється, клавіша "A" відповідає за розворот символу проти годинникової стрілки на крок 22,5°, а "D" – за розворот символу за годинниковою стрілкою на крок 22,5°, тобто поворот символу здійснюється шляхом послідовного завантаження з БД символів відповідного зображення, поверненого на певний кут. Кадр динамічної сцени показано на рис. 4.

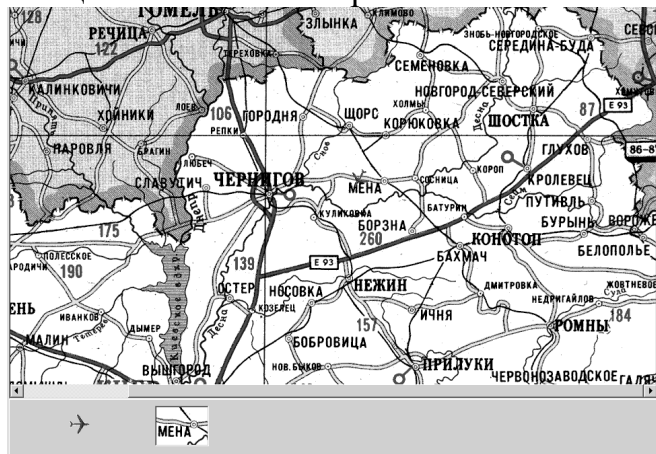


Рис. 4. Інтерфейс програми, яка моделює динамічну сцену.

Вікно відображення картографічної інформації, яка представлена так само, як і символ літака, в растровому вигляді, не вміщує в себе цілковитого картографічного зображення спостережуваної ділянки місцевості. Тому при «підході» символу до меж вікна необхідно пересувати зображення карти в напрямку, що відповідає курсу польоту літака. Для цього в програмі реалізовано процес скролінгу карти. А для візуалізації ділянки карти, що знаходиться під летючим об'єктом, виділяється прямокутне вікно, в якому відображається ділянка місцевості під символом і навкруги.

При організації «динаміки» символів враховується реальна швидкість переміщення об'єктів, що впливає на значення екранної

швидкості символу об'єкту, яка ще залежить й від масштабу карти, що виступає в якості фону. В нашому випадку екранна швидкість об'єкту є функцією залежності від кроку пересування символу на нове місце та від часу перемальовування символу в цій позиції. Щоб досягти плавного переміщення символу по карті, крок пересування символу вибирається мінімальним, а час його перемальовування встановлюється залежно від того, швидко або повільно потрібно переміщати символ.

Слід зазначити, що для деяких застосувань (при використанні карти дрібного масштабу, що охоплює велику територію, для моніторингу ситуації на ділянці місцевості) зручно відображати об'єкти, що рухаються, у вигляді простих символів (матриця розміром до 8x8 пікселів). При візуалізації повітряної обстановки зображення символу об'єкту характеризується двома координатами (x,y) : з прив'язкою до карти такими координатами є географічні – широта і довгота, що одержуються, наприклад, за допомогою GPS-приймачів, до яких додається третя – часова (t) .

Для побудови відносно простих зображень кадру динамічної сцени, яка включає прості відомі символи об'єктів, не треба забувати наробки „старих ” вчених. Пропонується наступна програма, що входить в розроблений нами пакет, в якій на фоні статичного растрового зображення моделюється лінійне переміщення об'єктів, представлених векторними символами у вигляді умовних знаків – хреста і кола.

Побудова символів векторним способом дозволяє спростити процес їх перетворення (зсув, поворот, масштабування), що значно скорочує обсяг використовуваних обчислювальних ресурсів (в першу чергу об'єму пам'яті) на відміну від використання растрових символів. Для цього використано відомі алгоритми: ЦДА, Брезенхема і Мічнера.

Нехай дано зображення символу-хрестика, яке будується попиксельно за алгоритмами ЦДА і Брезенхема для побудови ліній таким чином: по дві точки вліво, вправо, вгору і вниз від серединної точки. Припустимо, що його початкове положення на карті, крапка А, рис. 5, задано координатами серединної точки (x_1, y_1) . Щоб перемістити все зображення хреста в точку Б, треба привласнити координати серединній точці – (x_2, y_2) , тобто здійснити паралельне перенесення всього зображення в точку Б, $x_2 = x_1 + a$, $y_2 = y_1 + b$, де: a, b – прирости координат відповідно по осях X і Y .

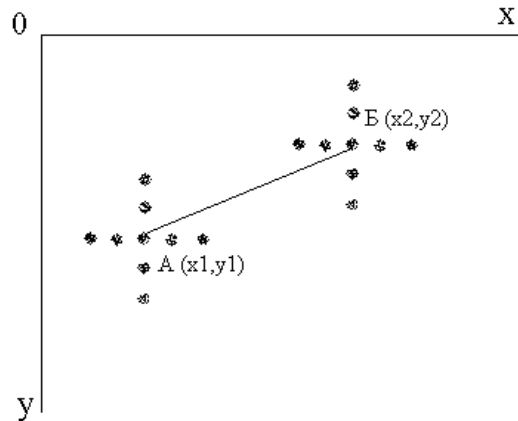


Рис. 5. Ілюстрація переміщення символу-хреста.

Відстань від однієї крапки до іншої c розраховується за формулою: $c = \sqrt{a^2 + b^2}$.

При переході від однієї позиції до іншої зображення хреста на попередній позиції стирається.

Приріст координат вибирається відносно невеликим з метою уникнення різких стрибків і забезпечення плавності переміщення символу, адже збільшення швидкості руху за рахунок збільшення кроку переміщення призводить до стрибкоподібного переміщення символу, при якому відбувається мерехтіння символу, що, в свою чергу, призводить до стомлення оператора, завдяки виникаючому навантаженню на очі. Тому можна поступити так: крок переміщення вибрати таким, щоб він відповідав критерію адекватного сприйняття оператором плавного переміщення (час виведення зображення на екран лімітується часом побудови кадру) і потрібно зменшити час «простоявання» (затримки) символу в конкретній точці. Якщо частота оперативно-запам'ятовуючого пристрою регенерації в системі реального часу рівна 50 кадрам в секунду, час побудови відеофайлу кадру буде дорівнювати 20 мс.

Нижче наводиться алгоритм лінійного переміщення зображення хреста, рис. 6.

Аналогічним чином відбувається й лінійне переміщення символу, представленого колом з координатами центру (x_0, y_0) та радіусом r_0 , причому, з погляду часових витрат, «розмір» зображення символу не робить істотного впливу на організацію процесу переміщення. x, y – координати центру «нового кола», r – його радіус.

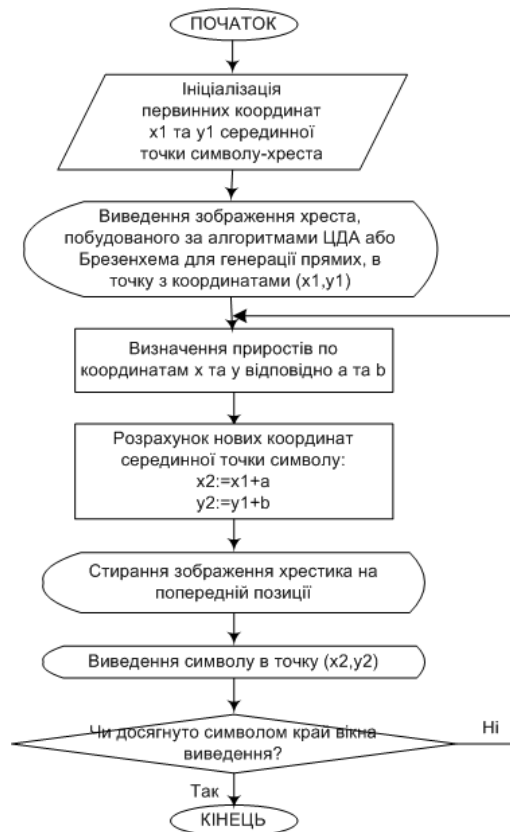


Рис. 6. Алгоритм лінійного переміщення символу-хреста.

Приведемо алгоритм рішення задачі переміщення кола:

1. Задаються початкові значення змінних $x0$, $y0$, $r0$.
2. Малюється коло по заданим в п.1 параметрам. Генерація кола може відбуватися згідно алгоритму Брезенхема або по його модифікації – алгоритму Мічнера.
3. Обчислюються координати центру “нового кола” (x,y) , як правило, радіус r ініціалізується значенням, рівним $r0$.
4. Стирається “старе коло”.
5. Замінюються параметри “старого кола” новими значеннями, тобто $x0:=x$, $y0:=y$, $r0:=r$ та відбувається перехід до пункту 2. Можливий інший варіант: малюється “нове коло” по параметрах, заданих в п.3 та відбувається перехід до пункту 3.

В пункті 5 алгоритму відбувається оновлення змінних $x0$, $y0$, $r0$. Їм привласнюються значення x , y , r , і з цієї миті коло стає “старим”. При наступному проході програма оперуватиме з новими значеннями параметрів “старого кола”.

Для реалізації лінійного переміщення хреста, вписаного в коло, «об'єднуємо» вище описані міркування для побудови хреста і кола. В результаті маємо наступну програму:

uses graph, crt;

```

var grdriver, grmode, errcode:integer;
    grpath:string;
x,y,r:integer;
color, bkcolor:word;
begin
grdriver:=vga;
grmode:=vgahi; {640x480}
grpath:='c:\bp\bgi';
initgraph(grdriver, grmode, grpath);
errcode:= graphresult;
if errcode <> grok then halt(1);
x:=10; y:=200; r:=55;
setbkcolor(blue);
bkcolor:=getbkcolor;
repeat
color:=lightgray;
setcolor(color);
circle(x,y,r);
line(x-r,y,x+r,y);
line(x,y-r,x,y+r);
delay(10000);
setcolor(bkcolor);
circle(x,y,r);
line(x-r,y,x+r,y);
line(x,y-r,x,y+r);
putpixel(x,y,color); {sled}
x:=x+2;
until (x>500);
setcolor(red);
outtextxy(10,10,'The movement is completed!');
readln;
closegraph;
end.

```

Результат роботи програмного додатку по лінійному переміщенню векторних символів на растровому статичному фоні показано на рис. 7.

Висновки. Розроблений ППП візуалізації повітряної обстановки у вигляді динамічних сцен, які являють собою накладувані на картографічний фон символи рухомих об'єктів, може використовуватися в реально діючих НГК РЧ.

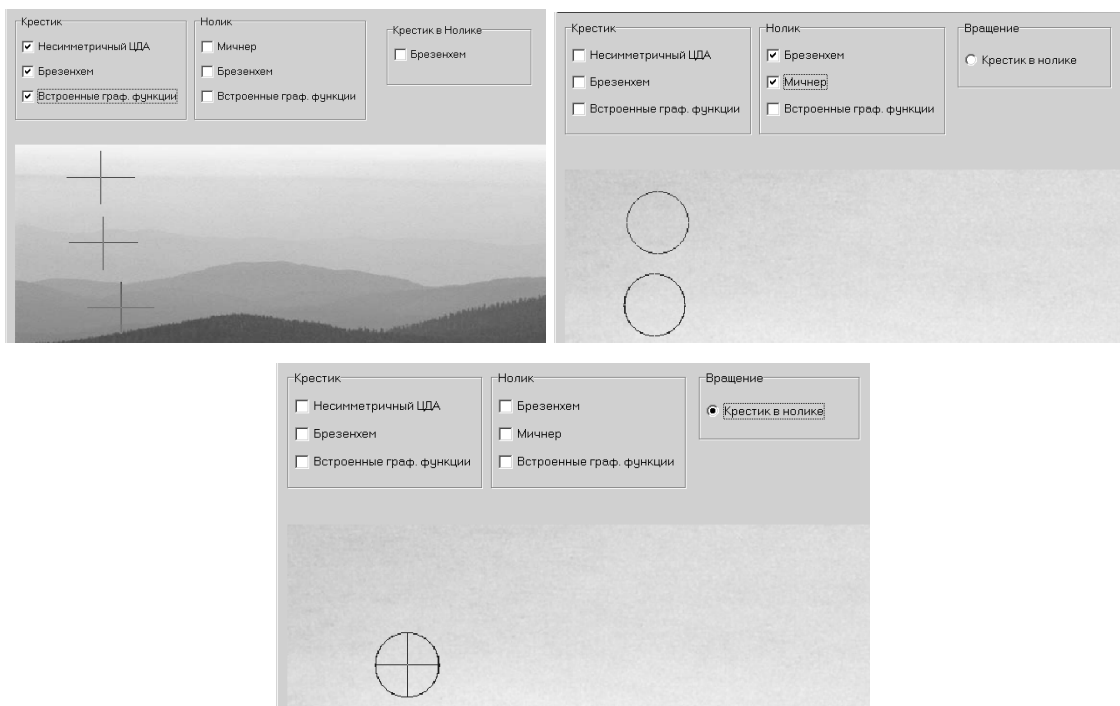


Рис. 7. Кадри лінійного переміщення векторних символів на статичному растровому фоні.

Література

1. Маначинский А. Информационное оружие: Мифы и реальность // «Посредник». – 1997. - №5 (494).
2. Айзекс Р. Дифференциальные игры. - М.: Мир, 1967. - 480 с.
3. Понтрягин Л.С. Избранные научные труды. – М.: Наука, 1988. - т.2. - 576 с.
4. Красовский Н.Н. Игровые задачи о встрече движений. - М.: Наука, 1970. - 420 с.
5. Батухтин В.Д. Экстремальное прицеливание в нелинейной игре сближения, ДАН СССР, 1972. - №1. – 207с.
6. Пшеничный Б.Н. Простое преследование несколькими объектами / Кибернетика. - 1976. - № 3. - С.145-146.
7. Чикрий А.А. Конфликтно управляемые процессы. - К.: Наукова думка, 1992. -384 с.
8. Жуковский В.И., Чикрий А.А. Линейно-квадратичные дифференциальные игры. - К.: Наукова думка, 1994. - 320 с.
9. Васюхин М.И. Метод разрешающих функций при решении задач преследования в интерактивных геоинформационных комплексах оперативного взаимодействия // Кибернетика и вычислительная техника. - К., 2001. - Вып.132.- С.21-32.
10. Васюхин М.И., Капштык О.И., Касим А.М. Алгоритмы построения зрительных сцен в аэронавигационных геоинформационных системах реального времени // УСИМ. – 2008. - №3(215). – С.79-84.

Отримано 24.04.09