

УДК 004

*Купрієнко М.С., здобувач,
Варганова Д.О., ст. викладач
Державний університет «Житомирська політехніка»*

МОДЕЛЮВАННЯ ПЕРЕТИНУ ДВОХ ПРЯМИХ У ПРОСТОРІ ЗАСОБАМИ ПРОГРАМУВАННЯ

Проблеми просторової геометрії лежать в основі багатьох сучасних технологій та наукових досліджень. Зокрема, у таких сферах як комп'ютерна графіка, робототехніка, мехатроніка, системи автоматизованого проектування (САПР). В багатьох задачах важливим завданням є визначення взаємного положення об'єктів у тривимірному просторі одне відносно одного.

Однією з головних цілей є визначення взаємного розташування прямих у тривимірному просторі, а саме чи перетинаються вони, паралельні, чи є мимобіжними, або збігаються. Якщо прямі перетинаються, то потрібним є визначення координат точки перетину для подальших обчислень та побудов.

Математична модель:

Параметричне рівняння, яке задає координати всіх точок прямої у тривимірному просторі:

$$x = l \cdot t + x_0; y = m \cdot t + y_0; z = n \cdot t + z_0,$$

де (x_0, y_0, z_0) – координати точки, що лежить на прямій:

$\{l; m; n\}$ – координати напрямного вектора прямої.

Розглянемо дві прямі, які задані параметрично:

Пряма L_1 : $x_1 = l_1 \cdot t + x_{01}$; $y_1 = m_1 \cdot t + y_{01}$; $z_1 = n_1 \cdot t + z_{01}$.

Пряма L_2 : $x_2 = l_2 \cdot s + x_{02}$; $y_2 = m_2 \cdot s + y_{02}$; $z_2 = n_2 \cdot s + z_{02}$.

Точка перетину цих прямих повинна задовольняти дві системи рівнянь одночасно, що утворює систему трьох рівнянь з двома невідомими і змінними t і s [1].

$$l_1 \cdot t + x_{01} = l_2 \cdot s + x_{02};$$

$$m_1 \cdot t + y_{01} = m_2 \cdot s + y_{02};$$

$$n_1 \cdot t + z_{01} = n_2 \cdot s + z_{02}.$$

Розглянемо алгоритм розв'язку:

1. Ініціалізація та вхідні дані. Користувач задає координати точок та компоненти напрямних векторів для двох прямих.

2. Перевірка на паралельність. Здійснюється перевірка на колінеарність векторів за допомогою порівняння їхніх векторних добутоків. Обчислюється векторний добуток напрямних векторів. Якщо його довжина близька до нуля (з урахуванням заданої похибки $EPS = 1e-9$), вектори вважаються паралельними. Якщо вектори паралельні,

алгоритм перевіряє, чи належить точка першої прямої другій прямій. Якщо істина – прямі збігаються, якщо ні – паралельні.

3. Верифікація. Знайдені параметри t і s підставляються в третє рівняння. Якщо воно також справджується, прямі перетинаються. Знаходимо значення координати z для кожної з прямих в знайдених точках. Якщо різниця між цими значеннями не перевищує задану точність EPS, прямі перетинаються.

4. Пошук точки перетину. Якщо вектори не паралельні, система розв'язується відносно параметрів t і s за методом Крамера, використовуючи перші два рівняння. Таким чином знаходимо точки перетину даних прямих.

5. Визначення мимобіжності. Якщо на етапі верифікації рівність для z -координати не виконується, прямі вважаємо мимобіжними [2].

Фрагмент реалізації програми на мові C (частина задачі, коли вже знайдено, що прямі перетинаються, знаходження координат точки перетину):

```
if (fabs(z1_check - z2_check) < EPS)
{
    double x_int = l1 * t + x01;
    double y_int = m1 * t + y01;
    double z_int = n1 * t + z01;
    printf("Точка перетину: (%.2f, %.2f, %.2f)\n", x_int, y_int, z_int);
}
```

Отже, реалізація даного алгоритму надає можливість ефективно зрозуміти взаємне розташування двох прямих у тривимірному просторі. Розроблений програмний модуль є універсальним інструментом, який можна використовувати не лише для базових геометричних задач, а й може бути інтегрований у складніші обчислення для розв'язання прикладних задач у робототехніці, комп'ютерній графіці та архітектурі, інших галузях, де необхідний просторовий аналіз.

Список використаних джерел:

1. Onlinemschool: Рівняння прямої [Електронний ресурс] – Режим доступу до ресурсу:

https://ua.onlinemschool.com/math/library/analytic_geometry/line/.

2. GeeksforGeeks: How to check if two given line segments intersect? [Електронний ресурс] – Режим доступу до ресурсу:

<https://www.geeksforgeeks.org/dsa/check-if-two-given-line-segments-intersect/>.