

УДК 681.3.082.5

С.О. Нестеренко, канд. техн. наук, доцент
Чернігівський державний технологічний університет, м.Чернігів, Україна

ПОБУДОВА ПРОСТОРОВОЇ ЛІНІЇ ПЕРЕТИНУ ПОЛІГОНАЛЬНИХ СІТОК

Стаття містить аналітичний виклад головних кроків алгоритму побудови просторової ламаної лінії перетину двох полігональних сіток, заданих в уявленні Б. Баумгарта.

Постановка проблеми. Важливим розділом комп'ютерної графіки є комп'ютерне геометричне моделювання (CAGM – Computer-Aided Geometric Modeling), мета якого полягає в створенні моделей різних геометричних об'єктів – як двовимірних, так і тривимірних. Одним з найпоширеніших прийомів створення комп'ютерних моделей тривимірних об'єктів [1] є спосіб конструктивного геометричного моделювання суцільних тіл (CSM – Constructive Solid Modeling), в якому більш складні об'єкти утворюються з більш простих тривимірних примітивів (блоків, сфер, конусів, тороїдів тощо). Базовий метод такого роду конструювання полягає у виконанні над парами об'єктів теоретико-множинних (бульових) операцій [1] знаходження об'єднань, перетинів та різниці останніх.

Оскільки тривимірні об'єкти апроксимуються найчастіше полігональними сітками [2] (скорочено “полісітками”) з плоских трикутних та чотирикутних граней, то конструювання тривимірних об'єктів за допомогою теоретико-множинних операцій має підтримуватись адекватними операціями над згаданими полісітками. Коли пара сіток частково суміщається в просторі, то знаходження результату будь-якої бульової операції вимагатиме побудови просторової ламаної лінії перетину (ПЛЛП), яка є спільною для поверхонь обох полісіток (рис. 1). Об'єкт, який буде результатом об'єднання, перетину або різниці суцільних тіл, що моделюються полісітками, включатиме побудовану просторову лінію до своєї моделі як складову частину множини його ребер.

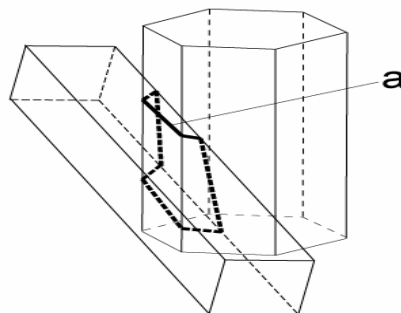


Рис. 1. Композиція двох полісіток: *a* – просторова лінія перетину полісіток

Побудова такої просторової лінії не є простим завданням з точки зору побудови коректного та швидкого алгоритму її вирішення. Ця стаття присвячена опису базових алгоритмічних кроків для побудови ПЛЛП з викладенням прийомів прискорення перебірних операцій, які зустрічаються в алгоритмі.

Мета статті. Мета роботи полягає в формулюванні математичних і алгоритмічних принципів вирішення складної задачі – побудови просторової ламаної лінії перетину двох полісіток.

Опис полісітки суцільного тіла

Довільна полісітка може бути описаною, як:

$$M = (V, E, F),$$

де V , E , F – множини вершин, ребер та граней відповідно.

Множина вершин визначається як

$$V = \{V_i(x_i, y_i, z_i) | i \in (1..n)\}, \quad (1)$$

де x, y, z – декартові координати вершини; i – номер вершини; n – кількість вершин.

Множина ребер визначається як

$$E = \{E_k(V_i, V_j) \mid (V_i, V_j) \in V, k \in (1..m), (i, j) \in (1..n)\}, \tag{2}$$

де m – кількість ребер полісітки.

Множина граней визначається як

$$F = \{F_i(\{E_p\}) \mid \{E_p\} \subset E, p \in (1..t_i)\}, \tag{3, a}$$

де t_i – кількість ребер, які є контуром i -ї грані.

Формула (3, a) визначає кожну грань як ланцюжок ребер, що зустрічаються при стандартному [3] обході грані за годинниковою стрілкою. Між тим, зустрічаються ситуації, де доводиться використовувати опис грані через множину її вершин у порядку обходу за годинниковою стрілкою:

$$F = \{F_i(\{V_p\}) \mid \{V_p\} \subset V, p \in (1..t_i)\}. \tag{3, б}$$

Наведений опис полісітки не віддзеркалює в повній мірі топологічну інформацію про неї. У зв'язку з цим до наведеного часто додають ще таблицю ребер у так званому «крилатому» уявленні Б. Баумгарта [3], яке пояснюється на рисунку 2. На рисунку показане ребро a , яке проходить від вершини X до вершини Y ; номерами 1 та 2 позначені грані, які розташовані відповідно зліва та справа від ребра; ребра b та d являють собою відповідно ребро-попередник та ребро-послідовник ребра a при обході лівої грані 1 за годинниковою стрілкою; така ж роль ребер e та c , але при обході правої грані 2.

Таблиця 1

Формат таблиці ребер для «крилатого» уявлення (Баумгарта) полісітки

Ребро	Вершини		Грані		Обхід зліва		Обхід справа	
	Початок	Кінець	Зліва	Справа	Перед	Після	Перед	Після
a	X	Y	1	2	b	d	e	c

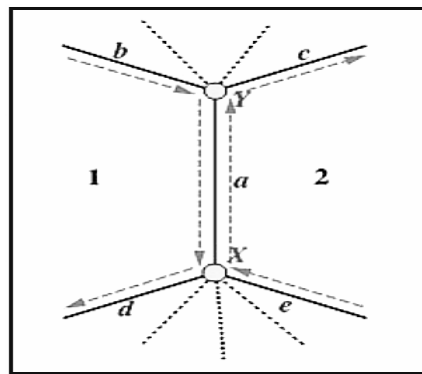


Рис. 2. Ребро полісітки в «крилатому» уявленні

Структури даних за виразами (1)-(3) та таблиця Баумгарта є достатніми для маніпулювання даними в границях цілей, які були визначені вище.

Визначення пар граней, які гарантовано не перетинаються. Перша проблема, з якою зустрічається розробник програмного застосування для побудови ПЛЛП, є прискорення виконання попарного перебору граней полісіток, що перетинаються на предмет визначення, чи вони дійсно перетинаються чи ні. Реальні полісітки нерідко складаються з великої кількості граней. Очевидно, що загальна кількість розрахунків тут буде пропорційна $N \cdot M$, де N та M – це кількості граней обох полісіток.

Визначення факту перетину двох непаралельних граней, обмежених багатокутниками G_1 та G_2 полягає у виконанні такої послідовності дій:

1. Визначити просторову пряму L – лінію перетину площин граней. При цьому і площини граней, і шукана лінія розглядаються як безграничні.

2. Пошук точок перетину L з G_1 та G_2 . Навіть коли для кожного полігона G_1 та G_2 пари таких точок $PG_{1,2}$ та $PG_{2,2}$ знайдені, фактичний перетин граней матиме місце лише тоді, коли відрізки $(PG_{1,1} - PG_{1,2})$ та $(PG_{2,1} - PG_{2,2})$ матимуть спільну частину, яка, власне, і є новим ребром $E_{G_1-G_2}$, спільним для G_1 та G_2 . Очевидно, що визначення $E_{G_1-G_2}$ є предметом додаткового аналізу.

З вищевикладеного зрозуміло, що примітивний підхід, який може полягати у виконанні цих дій повним перебором усіх пар граней двох тривимірних об'єктів, тобто $N \cdot M$ разів, небажаний з точки зору трудомісткості такого алгоритму. Дії (1) та (2) треба виконувати лише для таких пар граней, котрі не відносяться до таких, які гарантовано не пересікаються.

У зв'язку з цим виглядає раціональним попередньо вдатися до швидкого аналізу взаємного розташування граней у просторі за допомогою так званих «оболонки» граней [1].

«Оболонкою» H грані назвемо габаритний прямокутний паралелепіпед, в який вписується грань. Він описується шістькою максимальних та мінімальних координат вершин грані $H = H(x_{\min}, x_{\max}, y_{\min}, y_{\max}, z_{\min}, z_{\max})$. Програмне використання оболонок потребує включення такої структури до опису кожної грані.

Швидка детекція пари граней, які гарантовано не перетинаються, у своїй основі має те, що їх оболонки теж не перетинаються. Але останній факт встановити значно легше, бо він зводиться до простого порівняння координат і не вимагає об'ємних обчислень за пунктами (1) і (2), описаними вище.

Основною структурою даних, яка має бути одержаною в результаті такого аналізу, є матриця перетинів:

$$M_c = \begin{bmatrix} m_{11} & m_{11} & \dots & m_{1M} \\ m_{21} & m_{22} & \dots & m_{2M} \\ \dots & \dots & \dots & \dots \\ m_{N1} & m_{N2} & \dots & m_{NM} \end{bmatrix}. \quad (4)$$

Кожна компонента цієї матриці дорівнює

$$m_{ij} = \begin{cases} 1, & \text{якщо оболонка грані } i \text{ перетинається з оболонкою грані } j; \\ 0 & \text{у протилежному випадку.} \end{cases}$$

Якщо оболонки в тривимірному просторі перетинаються, то перетинаються також їх проекції на координатні осі Ox , Oy , Oz . Якщо хоча б одна пара проекцій оболонок першого і другого об'єкта не перетинається хоча б уздовж однієї координатної осі, то і самі оболонки не перетинаються. Тому процес заповнення матриці (4) має проходити в три кроки, відповідно до кількості декартових координат $(x, y$ та $z)$. На кожному кроці в матрицю M_c вносяться нулі для тих пар індексів оболонок, проекції котрих на вісь, яка аналізується, не мають спільних ділянок. Якщо для деякої пари (i, j) на попередніх кроках був одержаний нуль, то на поточному кроці оболонки з цими індексами перевіряти не треба, вони вже не будуть перетинатись гарантовано.

Вважаємо відомими масиви оболонок усіх граней першого та другого об'єктів, що перетинаються:

$$MH_1 = \{H_{1i} \mid i \in 1 \dots N\}; \\ MH_2 = \{H_{1j} \mid j \in 1 \dots M\}.$$

Процес заповнення матриці M_c можна описати C-подібним псевдокодом:

```
for(i=1; i++; i<N) for(j=1; j++; j<M) m[i][j] = 1; // ініціалізація матриці Mc
// визначення неперетинань вздовж Ox
for(i=1; i++; i<N) {
```

```

for(j=1; j++; j<M){
  if(m[i][j] = 1){ // раніше факт перетину не був виявлений
    if (!((xmin2[j]<xmax1[i]) && (xmax2[j]>=xmax1[i]) | |
      ((xmin2[j]<xmin1[i]) && (xmax2[j]>=xmin1[i])))
      m[i][j] = 0; // пара [i][j] не перетинається }}.

```

Потім ті ж самі дії повторюються для координат y та z . У результаті ми одержимо сильно проріджену матрицю M_c . Подальшому аналізу слід піддавати тільки ті пари граней, для яких $m_{ij} = 1$, що суттєво зменшує загальний обсяг роботи. При цьому слід зауважити, що факт не перетинання оболонок є гарантією не перетину самих граней, але зворотне твердження несправедливе. Остаточна відповідь на питання «чи дійсно перетинаються грані» дається на подальших кроках описуваного алгоритму.

Визначення загальної лінії двох граней.

Вважатимемо для конкретності, що розглядається пара непаралельних плоских граней F_1 та F_2 , і для кожної з них обчислене рівняння площини:

$a_1*x + b_1*y + c_1*z + d_1 = 0$ – для грані F_1 та $a_2*x + b_2*y + c_2*z + d_2 = 0$ – для грані F_2 . (5)

Знаходження просторової прямої зводиться [3] до визначення, по-перше, направляючого вектора прямої і, по-друге, точки, що належить цій прямій.

Направляючий вектор знаходиться [5] за формулою:

$$\bar{D} = \begin{vmatrix} \bar{i} & \bar{j} & \bar{k} \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{vmatrix} = \bar{i}D_x + \bar{j}D_y + \bar{k}D_z. \quad (6)$$

Опорну точку (x_0, y_0, z_0) прямої знаходять [5] як точку її перетину з якоюсь з координатних площин XOY , XOZ або YOZ . Для цього призначають нульове значення одній з трьох координат x_0, y_0, z_0 , а дві інших заходять як рішення лінійної системи з двох рівнянь, як показано нижче. Правильне призначення початкової нульової координати вимагає попередньої перевірки положення вектора D на можливу паралельність координатним осям. Можливі варіанти згаданого положення наведені в таблиці 2.

Таблиця 2

Варіанти розташування вектора D відносно координатних осей

№	D_x	D_y	D_z	Призначити нульове значення	Знайти як рішення системи рівнянь
1	0	≠0	≠0	y_0 (або z_0)	x_0, z_0 (або y_0, z_0)
2	≠0	0	≠0	x_0 (або z_0)	y_0, z_0 (або x_0, y_0)
3	≠0	≠0	0	x_0 (або y_0)	y_0, z_0 (або x_0, z_0)
4	0	0	≠0	z_0	x_0, y_0
5	0	≠0	0	y_0	x_0, z_0
6	≠0	0	0	x_0	y_0, z_0
7	≠0	≠0	≠0	Будь-який із варіантів 1-6	

Пояснимо вищезгадане на прикладі варіанта 6. Покладаємо $x_0 = 0$. Значення y_0 та z_0 знайдемо з системи рівнянь

$$\begin{aligned} a_1*x_0 + b_1*y_0 + c_1*z_0 + d_1 &= 0; \\ a_2*x_0 + b_2*y_0 + c_2*z_0 + d_2 &= 0. \end{aligned}$$

Пропускаючи тривіальні алгебраїчні перетворення, одержуємо:

$$y_0 = \frac{c_1 d_2 - c_2 d_1}{b_1 c_2 - b_2 c_1}, \quad z_0 = \frac{b_2 d_1 - b_1 d_2}{b_1 c_2 - b_2 c_1}.$$

Таким чином буде визначена безкінечна пряма, яка достовірно проходить через точку $P_0(x_0, y_0, z_0)$ в напрямку вектора $D(D_x, D_y, D_z)$. Довільну точку P на цій прямій можна знайти, використовуючи скалярний параметр t :

$$P = P_0 + tD. \quad (7)$$

Пряма за рівнянням (7) є спільною прямою для двох безграничних непаралельних площин з рівняннями (5). Де на цій прямій розташоване ребро дійсного перетину полігональних граней F_1 та F_2 аналізується нижче.

Визначення загального ребра двох граней.

Раніше ми висловлювали лише дві, причому досить слабкі, умови можливого перетину граней: по-перше, оболонки граней перетинаються, і, по-друге, площини граней не паралельні. Незважаючи на те, що ці умови є необхідними для існування спільного для граней ребра, вони не є достатніми. З рис. 3 видно, що достатніми умовами є ще додаткові такі:

У1. Спільна лінія площин граней (P_0, D) дійсно перетинає полігони обох граней;

У2. У випадку виконання умови (У1) відрізки $(x_{F1}-x_{F2})$ та $(x_{F3}-x_{F4})$ додатково мають спільну частину, тобто *ребро* $\leftrightarrow (x_{F1}-x_{F2}) \cap (x_{F3}-x_{F4})$.

Рис. 3, а показує випадок невиконання умови (У1), рис. 3, б ілюструє випадок виконання умови (У1), але не виконання умови (У2), а на рис. 3, в зображено випадок виконання обох умов, коли спільне ребро дійсно існує.

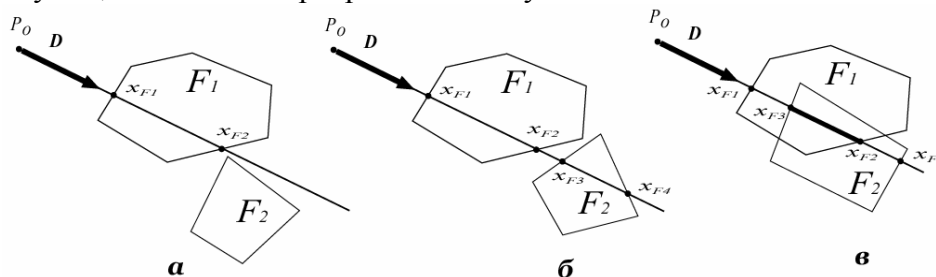


Рис. 3. Різні випадки розташування двох граней та спільної для них прямої:
 а, б – випадки, коли грані не мають спільного ребра; в – випадок, коли спільне ребро існує

Для визначення факторів, що забезпечують виконання умови (У1), нам знадобиться метод перерахунку координат вершин граней з «світової» системи координат (СК) $OXYZ$ систему координат, пов'язану з гранню $P_0X_FY_FZ_F$ (рис. 4).

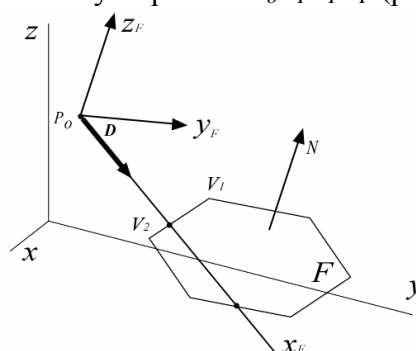


Рис. 4. Система координат грані $P_0X_FY_FZ_F$

Цю систему координат визначимо так. Вісь P_0X_F спрямована від P_0 вздовж напрямлюючого вектора D . Вісь P_0Z_F паралельна нормалі до грані F . Якщо описати базис СК грані як $\{\bar{i}_F; \bar{j}_F; \bar{k}_F\}$, то

$$\bar{i}_F = \frac{\bar{D}}{D}; \quad \bar{k}_F = \frac{\bar{N}}{N}; \quad \bar{j}_F = \bar{k}_F \times \bar{i}_F.$$

За цих умов СК грані є правою, як і світова СК $OXYZ$.

Перехід від світової СК до СК грані виконаємо за методикою [4], яка полягає у виконанні чотирьох кроків, у результаті яких СК OXYZ суміститься в просторі з СК $P_oX_FY_FZ_F$.

Крок 1. Суміщення початків координат, воно описується однорідною матрицею [5]:

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -P_{0x} & -P_{0y} & -P_{0z} & 1 \end{bmatrix}. \quad (8)$$

Крок 2. Обертання базису світової СК $\{i;j;k\}$ навкруги орту j з метою розташування ортів i та i_F в одній площині, перпендикулярній ортові k . Функції кута θ обертання

$$\cos \theta = \frac{i_{Fx}}{\sqrt{i_{Fx}^2 + i_{Fz}^2}}, \quad \sin \theta = \frac{i_{Fz}}{\sqrt{i_{Fx}^2 + i_{Fz}^2}}.$$

Матриця перетворення кроку 2 має вигляд

$$R_1 = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

На кроці 2 знаходимо й обчислюємо скореговані орти i_F та j_F , які знадобляться нижче (знак «штрих» означає «скорегований»):

$$\vec{i}'_F = \vec{i}_F R_1, \quad \vec{j}'_F = \vec{j}_F R_1.$$

Тут треба використовувати опис векторів в однорідних компонентах, з умовною четвертою компонентою, що дорівнює 1.

Крок 3. Обертання базису $\{i;j;k\}$ навкруги орту k на кут α з метою суміщення ортів i та i'_F . Функції кута обертання

$$\cos \alpha = i'_{Fx}, \quad \sin \alpha = i'_{Fy}.$$

Матриця перетворення кроку 3:

$$R_2 = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

На кроці 3 також обчислюємо $\vec{j}''_F = \vec{j}'_F R_2$.

Крок 4. Обертання базису $\{i;j;k\}$ навкруги орту $i = i'_F$ на кут β з метою суміщення ортів i та i'_F . Функції кута обертання

$$\cos \beta = j''_{Fz}, \quad \sin \beta = j''_{Fy}.$$

Матриця перетворення кроку 4:

$$R_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \beta & -\sin \beta & 0 \\ 0 & \sin \beta & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Загальна матриця переходу до СК грані

$$M = T_1 * R_1 * R_2 * R_3. \quad (9)$$

Тепер треба розрахувати для кожної вершини грані її координати (x_F, y_F, z_F) в СК грані. Зауважимо, що $z_F = 0$ та її обчислювати не треба.

Щоб швидко визначити ті ребра (припустимо без втрати загальності, що розглядається ребро $V_1 - V_2$) полігону грані, які перетинаються з віссю P_0X_F , використаємо критерій $y_{F1} \cdot y_{F2} < 0$, який практично означає, що кінці ребра знаходяться по різні боки осі P_0X_F . Якщо таких ребер у полігоні грані не визначено, то ми маємо випадок грані F_2 на рис. 3, а. Це означає, що дії із знаходження перетину поточної пари граней тут треба припинити і переходити до розгляду наступної пари граней відповідно до процесу обходу ненульових компонент матриці (4).

Після знаходження такого ребра координати точки перетину $(x_{PF}, 0, 0)$ знаходяться просто:

$$x_{PF} = x_{F1} + (x_{F2} - x_{F1}) \cdot y_{F1} / (y_{F1} - y_{F2}).$$

Знаходження координат кінців реального ребра, спільного для граней F_1 та F_2 (випадок на рис. 3, в) виконується після визначення x_{Fi} ($i=1...4$) звичайним порівнянням координат, тому тут не описується внаслідок простоти.

Включення нових ребер до моделі полісітки.

Коли нове ребро визначене, його треба занести до таблиці 1, а його кінці мають бути записані до множини (1). Перша дія є предметом досить складного топологічного аналізу, обговорення якого виходить за межі цієї статті. Друге потребує зворотного перерахунку координат кінців ребра з СК грані до світової СК за рівнянням

$$V_i = V_{iF} \cdot M^{-1},$$

де M^{-1} – матриця, зворотна до (9).

Очевидно, що матриця (9) буде матрицею загального виду, тому задача знаходження зворотної до неї не може бути вирішена спрощеними методами. Під час програмування цієї дії доведеться використовувати відповідний алгоритм зі стандартних бібліотек алгоритмів лінійної алгебри, хоча це дещо збільшує часові витрати на виконання обчислень.

Висновки.

1. Побудова просторової ламаної лінії перетину двох полісіток може бути виконана за регулярним алгоритмом, основні кроки якого викладені в статті.
2. Алгоритм містить у собі ряд переборів, які можна дещо прискорити шляхом попереднього виконання швидких порівнянь опуклих оболонок граней та ребер з метою уникнути детальних обчислень для пар примітивів, які гарантовано не перетинаються.
3. Алгоритм використовує як складові частини функції векторної та лінійної алгебри, тому для його програмної реалізації потрібно забезпечити доступ до відповідних математичних бібліотек.

Список використаних джерел

1. Фоли Д. Основы интерактивной машинной графики: в 2-х книгах. Кн. 2 / Д. Фоли, вэн Дэм А. – М.: Мир, 1985. – 368 с.
2. Шикин Е. Компьютерная графика. Полигональные модели / Е. Шикин, А. Боресков. – М.: Диалог-МИФИ, 2000. – 464 с.
3. Никулин Е. А. Компьютерная геометрия и алгоритмы машинной графики / Е. А. Никулин. – СПб.: БХВ-Петербург, 2003. – 560 с.
4. Роджерс Д. Математические основы машинной графики: пер. с англ. / Д. Роджерс, Д. Адамс. – М.: Машиностроение, 1980. – 240 с.
5. Справочник по математике для инженеров и учащихся вузов / И. Н. Бронштейн, К. А. Семенляев. – М.: Наука, 1981. – 720 с.